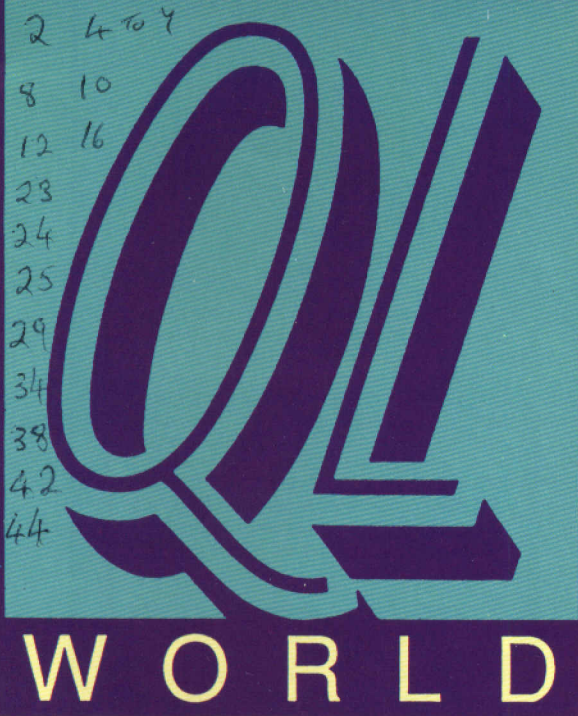


SINCLAIR

EVERY MONTH MARCH 1991 £1.75



**VALHALLA
REVISITED**
The Thor to date

YOUR PERSONAL GUIDE

Through the
computer jungle,
with a new
approach
to the QL
User Guide.

DBQL
All -SuperBasic
relational database

**QL
GUIDE**

SINCLAIR



Editor
Helen Armstrong

Production Controller
Jayne Penfold

Designer
Jeff Gurney

Advertising Manager
Jason Newman

Magazine Services
Yvonne Taylor

Advertising Production
Michelle Evans

Group Advertising Manager
Jean Dorza

Group Editor
John Taylor

Deputy Managing Director
Ray Lewis

Managing Director
Peter Welham

Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write The Editor, Open Channel, Trouble Shooter, or P'sion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.

Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.

Published by Maxwell Specialist Magazines, A Division of MCP Ltd., Sinclair QL World is distributed by IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS. Subscription information from: MSM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN.

UK: £21.00, Europe: £32.00, Rest of the World: £38.00.

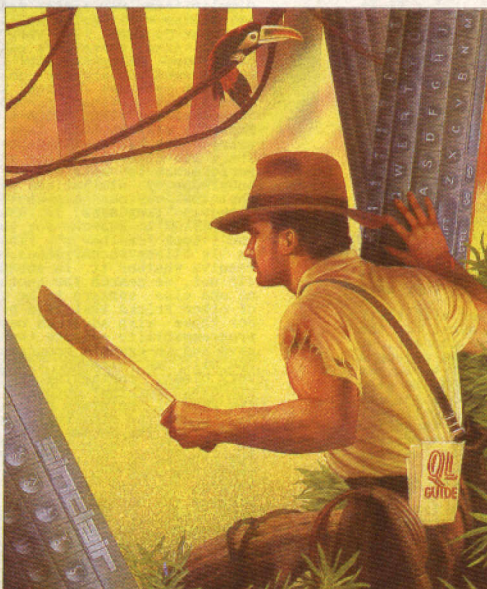
Airmail rates available on request. Typesetting by Ford Graphics, 8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657. Printing by Cradley Print. (0384) 411411. Sinclair QL World is published on the third Thursday preceding cover date.

© COPYRIGHT
SINCLAIR QL WORLD - 1991

CONTENTS

■ ■ MARCH 1991

- 9 **QL SCENE** ● Bristol meeting
- 10 **OPEN CHANNEL** ● Good suppliers
- 12 **TROUBLESHOOTER** ● Cheap and educational
- 15 **QL SCENE** ● Thor-desk now QTop
- 19 **VALHALLA REVISITED** ● The Thor so far
- 22 **PSION SOLUTIONS** ● Using Altkeys with Quill
- 25 **THE NEW QL USER GUIDE** ● File it under Sinclair
- 29 **QPAC1 AND THE POINTER ENVIRONMENT** ● User report
- 34 **LASER PRINTING** ● From text⁸⁷
- 37 **THE MICRODRIVE EXCHANGE** ● Shows its head
- 38 **DBQL** ● The all-SuperBasic database
- 42 **SOFTWARE FILE** ● QL Quick Posters
- 44 **PUBLIC DOMAIN SOFTWARE** ● Definitions and adventures
- 48 **SOFTWARE FILE** ● Data Design
- BACK COVER SUBSCRIPTION INFORMATION**



NEXT MONTH

ARCHED

A simple all-purpose line editor that runs within Archive

HARDWARE REVIEW

The Hewlett Packard DeskJet printer — a useful alternative to laser printing

NEW IN BRISTOL

A Quanta and QL workshop with a difference is to take place in Bristol in March.

The aim of this workshop, which will be advertised outside normal Quanta circles, is to attract QL users who are not Quanta members, as well as the regulars. "QL owners should stick together," said Quanta librarian Roy Brereton, one of the organisers.

The provisional schedule for the workshop is as follows: Saturday 9th March; opening 10am; 12 - 2pm SuperBasic for Beginners, and Machine Code (seminars); 3 - 5pm Advance SuperBasic; 5 - 7pm Trader demonstrations; 7 - 9pm Guest speaker (to be confirmed). Sunday 10th March; 11am - 1pm

Advanced SuperBasic; Psion Problems (clinic); 2 - 4pm Archive programming.

Speakers are expected to include Phil Borman, Jonathan Oakley, David Johnson, Hugh de Saram and Dennis Briggs. Refreshments will be available, and hot meals will be available at the hostel next door. The bar will be open at 'selected times'. There will be a bring-and-buy, and other activities apart from the listed seminars.

We hope to have this workshop covered by a *QL World* contributor.

The dates are 9th and 10th March 1991, and the venue is the Somerset Hall, Portishead, Bristol, "clearly signposted from Junction 19 of the M5".

DATADESIGN UPGRADE

The Van der Auweras have written to say that a minor bug in early versions of *The Painter*, mentioned by Simon Goodwin in *QL World* December 1990, has been fixed, and that "customers having any problems with any of our programs should contact us, because we cannot stand any bugs or mistakes in our programs." The Painter V4.01 "should be fully compatible with everything", but the Van der Auweras would again like to know.

The 'query' rate at QL World on The Painter and other VDA programs is one of the lowest, so we would say that they are living up to their aims.

A new version of *DATADesign*, the programmable QL database, is available from 1st March, V2.00, pro-

grammable from SuperBasic and machine code (so that you can create your own libraries for Pascal, C etc), includes improvements such as 'filter' (select a group of records to work with), 'duplicate' (make a second, almost-identical record without having to retype), and save all or parts of file, filtered, marked or selected records, and other updates.

The program comes with an extensive manual, and will cost 5000 Belgian Francs inc. p+p.

Progs/Van der Auwera is at Haachtstraat 92, 3020 Veltem, Belgium.

PDQL

Birmingham City Council Environmental Services Department Consumer Protection Division has contacted QL World to report the current situation regarding supplier PDQL.

The Enforcement Officer writes: "Two officers from this department have twice visited PDQL in an attempt to speak to Mr J Silk regarding various complaints received concerning his company. On both occasions no-one appeared to be present at the Camden Street address. This department then sent a recorded delivery letter to Mr Silk at PDQL, enclosing a list of 26 separate complaints, requesting assurances as to when these complaints will be satisfactorily resolved.

"It was requested that Mr Silk reply by 5th January 1991. To this date no reply has been received and as far as this department is aware the complaints are still ongoing."

Regrettably, it would appear that no new answers are forthcoming at the present. As far as *QL World* knows, PDQL is still resident at Camden Street. The Consumer Protection Division can still be contacted at **Birmingham Consumer Centre, 155-157 Corporation Street, Birmingham B4 6PH.**

The next All Formats Computer Fair is to be held at the New Horticultural Hall, Greycoat and Elverton Streets, Westminster, London from 10 am to 5pm on March 23rd 1991. Admission is £3 and the cost of a stand is low at £75. The "definitive bargain hunters' paradise in the computer industry" is still going strong. Phone 0225 868100 for further information about stands. Advance tickets are available from Mike Hayes, 8 Midgrove, Delph, Oldham OL3 5EL. Tel. 0457 875229.



OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide

somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

Good guys

After all the comments about suppliers letting down their customers, it is nice to be able to redress the balance a little with my experiences of both Digital Precision and Miracle Systems.

As a recent purchaser of Conqueror, I couldn't see how I could make MS-DOS boot up from my newly purchased 5.25 in drive, since the options the menu gave me were to load from flp1_ or flp2_ (both of which were already allocated to 3.5 in drives). I contacted the suppliers of both the software and the hardware on the telephone. Neither could sort me out in real time, but I was rung back by Digital

Precision to give me help and advice, to enable me to do what I wanted.

Special praise, though, must go to Miracle Systems, who actually sent me free of charge by return of post their Disk Adaptor Kit, to help me circumvent my problem. In my experience, this is typical of the service that they extend to their customers. A year or so ago, I made an enquiry about what rewiring I would need to do to make my old Centronics printer interface work from ser2. Miracle Systems' response was to send me a replacement ser1/ser2 Centronics printer interface at no cost to me – not even postage, which I offered to pay. And when I received my original Trump Card, there was a free Miracle Systems T-

shirt included in my order. This is certainly the way to treat your customers.

Michael Wood,
Knockholt,
Kent

Superbasic

It is with a great deal of regret that I have come to the conclusion that to upgrade my JSQL beyond the extra memory which is already attached, would not make financial sense. To add dual drives, a hard disc and replace my tv set with a decent monitor would cost little less than some quite powerful 80286 systems that are now on offer. For a comparatively small extra sum one could even contemplate upgrading to a 80386 system.

Personally I will retain my QL. I have grown very fond of it over a period of four years and it has been the means of entering a new and absorbingly interesting world. The thought of selling it is quite repugnant.

Your magazine has served me, and many others, well and there is a fair sized and essentially reliable service industry to meet the needs of the QL enthusiasts.

I am struck by the idea that there could be a bigger future for all concerned. I have looked at two or three other Basic languages and, although I may be biased, they do not have the fluency and the capabilities of SuperBasic. If Houses like Digital Research can emulate the PC environment to give compatibility with IBM's and the clones, why cannot they produce a package that would put QDOS and SuperBasic into PC systems? To be sure such a package would have to include the enhancements introduced by various toolkits and

Minerva.

If SuperBasic could be successfully promoted by such a method it would open up a whole new field for the sales of these software houses and their retail outlets which are already operating from a secure commercial base.

QL World, whose contents and articles are mainly concerned with SuperBasic, QDOS machine code and related software, could gain a subscriber market with a very good potential.

J W Hill
Matlock
Derbyshire

The idea may be a bit fanciful but I am sure that is worth some consideration. I will be the first purchaser.

Thor disks

Over the last few years I have made good use of my Thor, which I have expanded to include a hard disc of 20Mbytes capacity. Perhaps I have come to rely on it too much, as the QL is, after all, a fine computer and still streets ahead of its nearest competition, but I would very much like to add a second hard disk drive, at least.

I know that the Thor can cope with up to eight drives on its interface, but my problem is that I don't know what sort of hard disk is suitable. The one fitted was made by Rodime but nowhere have I seen a similar model on sale, nor have I seen anywhere the address of the Rodime company which, I understand, had severe financial difficulties at one time.

The components company RS do supply hard drives, one of which seems likely, the Seagate ST225, but while it looks suitable for an IBM-AT, I have no idea if it would work

Editor's notebook

The Quanta Workshop to be held in Bristol in early March is a meeting with a difference: it is not just for Quanta. Organiser Roy Brereton has said "QL owners should stick together". Meetings which bring diverse QL users together can only strengthen the QL. See *QL Scene* for details.

QL World's New User Guide, which starts in the centre pages this month, is both a new guide for users and a guide for new users. It responds to requests for clarifications of parts of the *Sinclair QL User Guide*, and is designed to lie alongside the Guide, not replace it. If you have a particular bugbear in the Guide, now is the time to let us know.

When I listed the Christmas Things that Went Astray, I missed one. A last minute system-crash left the *Editor's Notebook* apparently under the influence of the Christmas port, and a number of other small but confusing errors in the 'late bits'. The good news, for QL users anyway: it was £50,000 worth of dedicated page make-up kit that went down. Not the QL.

on my Thor. Is there someone still supplying a compatible drive, or do you have the address of Rodime?

I do need the extra disk space rather urgently and would rather not have to buy another computer just to give myself expandability—that seems too expensive an approach.

Edward Jones
Newton Longville
Bucks.

Editor's comment: We don't have the address of Rodime. Good Thor support is very difficult to come by. A word with Miracle Systems, whose phone number appears in their ad regularly in QL World, may be worth your while, as they have developed their own hard disk unit and might be able to give an opinion on your predicament.

QControl

I am interested in obtaining the QControl II board as described in the QL Connexions series circa 1987.

Is this board still available—if so, where from and how much does it cost?

If the board is no longer available, is there a kit or a set of circuit diagrams available?

I understand that some support microdrives, containing the source of the Qcontrol II rom and various tables of Qdos calls. Are these still available?

Are there any more details or descriptions of the actual Qcontrol II board hardware other than that published already in QL World?

I would like to suggest that a set of maths routines be generated to handle matrices (eg MATC=MATA*MATB but for addition, subtraction in verses, determinants and the like) for the DIY Toolkit series.

R P Kelly
8 Woodland Way
Stevenage
Herts SG2 8BT

Editor's Comment: Now you have me. The 1987 file is currently missing, the author, who I am fairly sure was Colin Opie, quit the QL community nearly three years ago to go industrial, and I do not know of any QL supplier who supports the Qcontrol board. It was a popular series at the time, and there must be a number of users who either have copies of the material, or possibly a board to sell. The

only information we had was as it appeared in QL World.

DMP graphics

I had the same problem as S. Curtis of Stoke Newington: namely that of getting a Centronics GLP dot matrix printer to print graphics. When graphics is being printed the Psion software leaves the spacing set as its default value of 1/6in instead of resetting it to the required value of 7/72in. This happens because the Psion programs were designed to work with the FX80 and compatible printers. The GLP requires a different spacing command.

The solution I devised is inelegant, in that it cannot be used from within the program, but it works.

Before loading and running the graphics program, switch on and set up the printer, and then send the following command from the keyboard:

OPEN \$ 3, SER1: PRINT \$ 3, CHRS (27); "1"; CLOSE \$ 3

If you omit to do the above before running your program, you will of course need to save any data and return to Basic before sending the command.

In the case of an unexpanded QL, the program may not return and it may be necessary to reset the computer and LRUN before proceeding.

A better solution would be to add a suitable piece of machine code in front of the graphics driver. Perhaps another reader would show us how to do this?

Dr David Scott
Lee Green
London SE12

Dump update

May I compliment Paul Walton on his screen dump utility in the September issue of QL World. However, readers with printers such as the Epson RX80+ may find that, although the printer roller rotates when the program is run, nothing is actually printed. This is because bit-image mode 5 called up by line 9180 is not supported by these printers.

To rectify this, amend line

HELP press F1	CURSOR move + with ↑ keys +	TEXT Insert: Type at New para: Press ENTER Delete: CTRL & ↑↓ Change mode: SHIFT & F4	TYPEFACE press F4	COMMANDS press F3 ESCAPE press ESC
------------------	--------------------------------------	---	----------------------	---

1.....2.....3.....4.....5.....6.....7.....8.....

LOAD Taskforce and MERGE the Dump procedure with it then RUN. Enter the name of the Psion program when prompted and when it is running use it in the normal way until you wish to save a screen. Then press (CTRL-C) to switch to SuperBasic and enter as a direct command

PAUSE 500 : dump <ENTER>

Immediately press (CTRL-C) to transfer back to Psion, followed by (SHIFT-F5) to redraw the screen. Then after a second or two you will get your printout. Don't use the keyboard while the printer is running or you may get some very peculiar printouts.

Most Psion screens use black paper and these look terrible when printed. The following short procedure reverses the colours leaving a narrow black border round the screen.

MODE: INSERT WORDS: 447 LINE: 48 PAGE: 1
TYPEFACE: Normal DOCUMENT: "psiondmp"

9180 by changing the CHR\$(5) at the end to either CHR\$(4) or CHR\$(6) when the program should run satisfactorily.

Having used Dump to copy a sample screen, I thought, what a pity the program can't be used to copy screens from the Psion suite. Then I thought a little harder, and realised that it can be, and from SuperBasic too, using Simon Goodwin's Taskforce from the April 1989 issue.

LOAD Taskforce and MERGE the DUMP procedure with it, then RUN. Enter the name of the Psion program when prompted, and when it is running use it in the normal way until you wish to save a screen. Then press CTRL C to switch to SuperBasic and enter as a direct command.

Then press CTRL C to transfer back to Psion, followed by SHIFT F5 to redraw the screen. Then, after a second or two, you will get your printout. Don't use the keyboard while the printer is running or you may get some very peculiar printouts.

Most Psion screens use black paper and these look terrible when printed. The following short procedure reverses the colours, leaving a narrow black border round the screen.

8000 DEFine PROCedure
PSIDUMP

8010 PAUSE 500:

REMark 10 seconds wait while you fiddle with the keyboard. Adjust to suit yourself.

8020 WINDOW 508,254,2,1

REMark Sized to leave narrow black border unaffected by RECOL

8030 RECOL 7,6,5,4,3,2,1,0:

REMark Most Psion screens print best with the colours reversed.

8040 dump

8050 END DEFine PSIDUMP

MERGE this with the dump procedure and to copy a screen:

1) Make sure your printer is switched on and loaded.

2) Press CTRL to switch to SuperBasic

3) Enter "PSIDUMP" and immediately ...

4) Press CTRL to switch back to Psion

5) Press SHIFT F5 to redraw the screen

6) Sit back and wait

The short lines in the top left hand corner of the printout are due to the flashing cursor, which will print or not according to whether it is on or off when that area of the screen is scanned.

J E Pearce
Bakersfields
Nottingham

T A R O U B L E

It's funny how answers to questions can turn up unbidden, ages after you've given up trying to find them. I've long been irritated by failure of the Trump Card screen dump function to reproduce red from the screen on my FX80-compatible printer, and to distinguish between green and black properly. Use of the parameters listed in the instructions defeated me, but I decided to have another try with them when some dumps to the laser printer were needed for a program review. Somewhere along the way, a dump appeared of a reasonable size and showing the green and red screen areas sensibly. The trouble was – as always – remembering what parameters had produced the success. It eventually came to light that the way to get the desired result was to ignore the parameters for the FX80 and use those for the MX80 instead. The line to put into the boot routine was:

SDP_KEY 'P':SDP_SET 1,3,0,0

The 1 indicates the MX80, the 3 gives the scaling to get a print about 18cm high by 20cm. wide, the first 0 is for *not* inverting the image (ie black comes out black), and the final 0 is for not using the random function to produce grey shades (the results using random shading were poor). The same line worked fine on the dot-matrix printer as well. Hopefully, the screen dumps accompanying my reviews will look a bit better in future (and it won't be necessary to fiddle with the RECOL command to try and get all the display reproduced).

ST Graphics

A point that should have been made here before concerns use of *Fleet Tactical Command* and other QL graphical programs on the Atari ST with the QL emulator. The Van der Auwera brothers from Belgium wrote and pointed out that the emulator includes a ZX8301 chip, which ensures that the ST screen is driven like the QL one, rather than by its normal driver, so there should be no problem running correctly-written programs, such as their own *The Painter*. They mentioned that a 4-colour, 768-by-280 pixel driver is being written for the ST emulator, and there is hope that the necessary modifications to QDOS will be made to support this enhanced screen mode (already being tested, according to later reports).

The QL is a bargain computer for anyone equipping a school or teaching facility, argues Bryan Davies, bringing one reader's experience to bear on the question.

Maybe, horses-for-courses. Why? Simply that a recent letter in the *Quanta* newsletter, from a schoolteacher, struck me as a very much more sensible comment upon the merits of the QL than do the (fairly frequent) derogatory remarks from some QL users about PCs. The teacher's implied point was that the educational establishment's policy of buying only certain brands of computer – notably the Research Machines' Nimbus and Acorn BBC and Archimedes – had inevitably lead to a monopoly market, and this has naturally lead to high prices.

Once Sinclair had been ruled out of the competition for the BBC-branded computer, the way was clear for 'non-competition'. In the case quoted in the teachers' letter, much persuasion had finally resulted in authorisation being granted to equip one school with QLs (from EEC Ltd.); the cost of 15 QLs was stated to be less than that of ONE Nimbus. As I read the figures, 15 Nimbus machines (14 of them being work stations) would have cost about £12,000. Presumably, an order for 15 QLs got the school a decent discount over the usual price, and one can assume the cost for them was less than £1,800.

Excellent

The obvious saving in money here should not be extended to arguing that the QL is – *per se* – a *better* computer than the Nimbus. The school wanted to teach pupils about computing, and the QL is an excellent computer for that purpose. Likewise, the Psion Quartet are excellent for teaching the basics of the main applications for which micros are used. Had the requirement been to train pupils for jobs in

offices, there would have not been much point in using QLs, since the application programs and the systems office workers are most likely to meet are centred on PCs.

Forget the PC

The main argument for a PC-type computer has to be commercial – maybe you are selling some service which requires the output to be from a PC program, or in PC disk format. If that argument doesn't apply to you personally, forget about buying a PC. It is not the type of computer to buy for fun. The QL is generally less trouble, more portable, much quieter, takes less space, and is more suitable for learning about programming and computing. It is very clear from the mail we see that many – perhaps most – QL owners do not require word-processing programs with more power than *Quill* provides, so why suffer the agony of getting to grips with the standard offerings on the PC? To some extent, one could say the same about *text*⁸⁷, and the new program *Perfection*, but they both offer a clear upgrade path from Quill, making it possible for the serious user to obtain many more features without the hassle of learning an entirely new 'language' of key combinations for invoking commands. In case you are not aware of the awful, non-friendly nature of command keying with PC programs, try these for starters:

Some Examples

Shift-F8 L S 0.5 (change line spacing to 12 lines per inch — *WordPerfect* 5),
Esc T L F1 [select with cursor] **Enter** (load a file — *Word* 5),
Ctrl-Q F [string] **Esc** (find a character string — *Wordstar* 3.4)

These are not particularly bad examples, but note the lack of consistency, and the combination of mnemonic keying with varying control keys. F1 is a standard key for invoking Help, so *WordPerfect* uses F3 instead, *MultiMate* uses Shift-F1. Esc is the standard key to back-up (escape) to the previous menu level, so *Word* uses it to display the command main menu (ie as F3 does in Quill), *Wordstar* uses it to initiate a search. When you want to get a document from disk and have it displaced

SHOOTER

M S O L V E D

on the screen, what word comes to mind? Load, retrieve, open? Word uses Transfer. No argument, there's some logic there, but who would think transfer? (The process is not a transfer anyway, since the original document remains on disk—Copy might make more sense.)

As one who has undergone more than his fair share of agony, getting to grips with numerous, incompatible command key-sets, I can only advise QL users to stick with the sensible structure that Psion provided. It was good in 1984, and is just as good now. Both *text*⁸⁷ and Perfection have commanded keyings based upon the Psion model, and learning them is not really difficult (although it does take plenty of time understanding some of the nomenclature). Equally important, *Software*⁸⁷ and Digital Precision are still enthusiastic about developing their QL programs further, and new features can be expected this year.

Minerva

It might appear to some readers that *QL World* contributors have been indulging in that standard journalistic practice of creating smoke where none exists, to provide material for articles during a period of time when nothing much is going on. However, you can rest assured that Helen, Mike, Simon and myself did not jointly agree to write about the Minerva QDOS-replacement, and make (some) cautionary comments about them. The fact is that QView seem to have been successful in distributing so many Minervas (or Minervae, to use Simon's form), and that had inevitably lead to a fair amount of discussion, revolving around the effects a Minerva has on QL operation.

We look at any development on the QL scene from appreciably different viewpoints, ranging from the largely non-technical one of the editor, through my own, mainly use-oriented attitude, to Mike's technical-but-commercial view, and then to Simon's very-detailed technical outlook. This is not to say we see things completely differently; we have all been made aware that Minerva is a subject needing discussion, but we tend to consider different aspects of its impact upon the QL scene. Asking for readers' comments produced the longest letter received by me on any subject to date—close on 8,000 words according to the

writer, Gerard Phelan. He feels that Minerva has been getting bad press, in *QL World* at least, and talks of 'mud slinging'.

No mud here

Nothing that I've read in *QL World* strikes me as warranting that description, so what is there we might agree on? Firstly, it seems definite that use of Minerva has resulted in changes in behaviour of some programs. The changes are not the same with all versions of Minerva; indeed, program problems apparent with one version may not show up with another. As is usual in such situations, more time seems to be spent justifying entrenched positions than fixing problem for the users. "Almost all incompatibilities discovered have been where software authors have ignored the QDOS Technical Manual" is a statement Gerard made, and one which invites a storm of criticism. Effectively, the statement is saying that the author(s) of the Minerva code are 'right', (almost) all the authors of programs which have trouble when used alongside (under?) Minerva are 'wrong'. One of the software writers to fall foul of a Minerva version is Tony Tebby; surely he is the one person who has previously been held up as The Authority on QDOS? It seems ironic that some of the coding for QJump products must come from a similar pen to that for Minerva.

What to do?

To return to the practical problem, what does the user do when faced with a program that no longer works correctly, once Minerva has been installed? Although no easy task, the temporary removal of Minerva and its replacement by the original QDOS would seem to be the simplest and most effective solution. It doesn't involve anyone other than the user; no need to write complaining letters to anyone, and hope for a solution from someone who doesn't agree it is *his* problem anyway. That is a rather negative approach.

If we take the other view—get either the Minerva *firmware* or the user program software modified—who is to be the judge upon which of the two courses must be taken? And who pays for the programming time required? Hopefully, ways of patch-

ing-around the various problems will be found, without great and financially demanding amounts of time having to be spent to discover them, and there will be sufficient goodwill between the parties involved to permit the work to be done, so that users do not suffer too much inconvenience. Surely, though, it is a bit late to come along with *firmware* changes to the operating system 6 or 7 years after the computer was launched?

Gerard didn't ask one question, but I'll give the answer anyway. There are two main reasons why I don't personally use Minerva. It would be unfair to writers of programs which are reviewed to introduce a potential source of conflict, if some trouble did occur; in any case, the system used for doing reviews needs to remain roughly similar to those of most QL users, to enable a representative view of program performance to be obtained.

Advantages

The other reason concerns the advantages to be gained by changing the operating system. My main QL has the JM version of QDOS; at one time, a JS was used, but that refused to work for reasonable uninterrupted sessions with my Trump Card and had to be put into the drawer. As far as I am aware, the JM causes me no significant trouble. Most of the listed plus-points for Minerva didn't mean much to me. They weren't related to any difficulties I'd experienced with user programs, so I couldn't judge whether or not any improvement was to be expected in these areas. The one point that did clearly relate to my use of the QL was that Minerva gives improved display speed for text; but not as much as *Lightning* does, and the latter was already in use on my system.

Hard disk

The letter was good in that it gave views on a variety of QL matters, and several of them need more space to discuss than is available within this column. Actually, more space was taken on hard disk use, and sub-directories, than on Minerva. Hard disk seems to me more important to system performance than any change in QDOS is likely to be. There were many versions of MS-DOS on the PC before

TROUBLE SHOOTER

hard disk came along, but I doubt if any of the changes to (the debugged) DOS had the impact that the arrival of hard disk did.

MS-DOS versions

Gerard incorrectly quotes the tendency of computer suppliers to sell MS-DOS 3.3 rather than 4 with PC systems as being because "it's not upwardly compatible", this apparently being the argument in support of Minerva. MS-DOS was avoided because it had (at least) one serious bug in it and would have caused suppliers untold trouble had it been supplied in its early form. In addition, version 3.3 worked fine for most users, so why change just for the sake of it? There is certainly a parallel there, between QL and PC, but it is limited; so far as I am aware, there has been no serious suggestion that existing PC user programs started giving trouble when used with DOS 4. The finger of suspicion pointed only at the operating system.

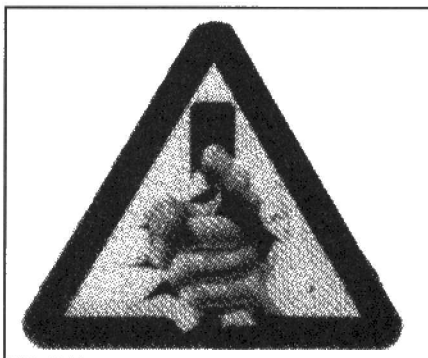
Laser printing was another point touched upon by Gerard. He noted that I may have given a false impression of the life of print drums. Early laser printers had drums with a quoted lifespan of 100,000 or more copies, but typical current types have drums of a throw-away type, for which the life is much less. The Epson GQ-5000 had a 'drum cartridge' with a claimed life of up to 20,000 pages. This unit cost typically £108 (all prices here exclude post and packing, and VAT), but it does include a 'collector unit' (see later). My first two toner cartridges have averaged 1,250 pages each, and they cost £16.50 each. The prints were primarily text, but there have been a fair number of graphics images printed also.

Longer life

There is yet another consumable item to allow for – a 'collector unit', which has a claimed life of up to 10,000 pages, and costs £59.00. You can find somewhat better prices than these, but it is advisable to stick to the approved items, at least during the guarantee period. Amortising the capital cost (excluding VAT) over five years, a yearly use of 10,000 sheets of 80 gsm paper works out to cost about 4p per printed page. In practice, more than 75% of the paper I use is scrap, and the page cost is more like 3.5p. On the same basis, my 9-pin dmp would work out to 1.5-2p per page, so I'm not unhappy with the size of the increase in cost from using the laser.

In the November *Open Channel* there were a few points which deserve comment. Julian Colomina Gonzalez wrote from Madrid, asking about networking. From what he says, he is trying to use the *Toolkit II* network facilities without fitting

plug-in Toolkit roms, or interfaces with TK2, to both QLs. My observation when I tried this was that you can't save a copy of the TK2 firmware from one QL and load it from disk on another and get the network to operate properly. You need the *hardware* in both QLs. The ways of dealing with overheating have been well documented, and many users may be able to solve nearly all of their lockup problems by the use of a cooling fan, but this won't help you if the source of the lockups is electrical interference, such as that caused by short term voltage fluctuations in the mains supply. To deal with this problem, you may need to get into the mains plugs on household appliances and fit suppression devices (varistors are my recommendation), and perhaps change the 5V voltage regulator in the QL, and/or fit a better external power supply.



Michael Hussman (of Hamburg) suggests that the QIMI interface is no longer available, now that QJump is no more. Can anyone (eg Care Electronics, Jochen Merz, Quanta) say if Tony Tebby did give anybody else the right to produce the QIMI? Back to the same subject, he suggests that – if Minerva is fitted – there is little point in buying *Lightning S.E.* rather than *SpeedScreen*. As the editor commented, time to get under the table before the shells start flying, after a comment like that! As a user of *Lightning* in its non-rom form, my thought is, why buy *any* of the other three? There is said to be reasonable compatibility between *SpeedScreen* and *Lightning* on one hand, and Minerva on the other, but the display speed gains from using any two or three together are not going to be great when compared to using just one, so is it worth risking complications?

No luck

There were some good programs in the PDQL adverts some months ago, but it doesn't presently look as though they are about to re-surface in any other supplier's adverts. In particular, there was Basic C-Port, the C compiler, *HardBack* and *Finder*. The first of these was a program for converting SuperBasic source code into C code, and there must be quite a few

QL users who would like to have that. There have been other C compilers available, but only one is regularly advertised now; the PDQL one appeared to be offering some things that programmers might want, but there were apparently difficulties associated with the use of existing code. *HardBack* is a back-up program for hard disk; anyone who has hard disk needs a back-up program, before disaster strikes (as it almost certainly will). *Finder* was what you might expect – a program to find elusive files, on a crowded hard disk. It would be good to see all of these programs on sale again.

Housekeeping

What would perhaps be even better, for hard disk users, is a simple housekeeping program capable of displaying both sub-directories and their files on the screen, simultaneously. Also needed is a file-unfragmentation program. You may not realise it with your files on floppy disk, but long files tend to get split up into several pieces and spread all over the disk; the more you can make changes to a file, the more fragmented it gets. This can become a real problem. Amongst the manifestations of the problem are a significant slowing-down of input/output operations. A recent comparison between two PCs emphasised the possible magnitude of the speed loss. A sample 'chapter' Ventura-speak for a document with several pages in it) was loaded into two computers which (nominally) differ in cpu speed by at least 50%; the machine which is clearly faster at displaying text or graphics on the screen was about ten times *slower* loading the chapter into *Ventura Publisher*. There could be other reasons for the difference, but file fragmentation looked to be the main one.

All-Formats

Judging by the lack of letters of complaint, the QL scene is settling down and the remaining suppliers are proving satisfactory (although one has not been advertising for a while). Maybe it is time for readers to write in with questions about problems which do not relate to suppliers, for a change.

Anyone looking for odd bits-and-pieces, such as cheap disks and PC-type keyboards, should consider going to one of the All-Formats computer fairs at the New Horticultural Hall in London. They have become an almost-monthly event; at first sight, this looked to be too frequent, but the public still come along in fair numbers and appear to be spending money despite the recession we read about. The number to call for the dates of fairs is (0457) 875229 (ask for Mike Hayes) or (0225) 447453 (John Riding).

QL

SCENE

COWO THOR-DESK IS RE-NAMED QTOP

Cowo Electronics' program Thor-Desk launched at the European Microfair in Brussels last October, has been re-named **QTop**. "This is necessary to point out that this user environment system is a real QL package, which goes in the same direction as QRam, QPac, QD, etc.", writes Urs Konig of Cowo.

QTop is billed as a user front end for all QDOS-compatible computers, in English, with manuals in English or German,

with a French version in preparation. TK Computerware are the agents in the UK, where QTop costs £35 on disk or £39 on mdv. In Germany, Qlympic Computer Systems are the agent, and supply the German version for DM99 on disk or DM109 on mdv. Cowo can supply QTop with manuals on disk at 95 Swiss Francs or mdv at 98 Swiss Francs. Please specify English or German.

Some features of QTop are:

written in size- and speed-optimised machine code to run on all QDOS-compatible systems such as QDOS itself, SMS-2, Minerva or Argos, including (a recent update) the unexpanded 128K QL and the Amiga running the QL emulator; completely multitasking; multi-windowing (except on unexpanded basic QLs); works in all display modes, including MODE 12 on the Thor XVI; single-key commands and menu bar control; supports all kinds of mouse systems; full DEVICE, FILE, JOB and MEMORY handling; access to foreign keyboard drivers, serial port translation tables, all CST- and Sinclair- (Psion quartet) supplied programs; full user supports and free updates, with many other advanced features.

An information sheet and order form is available from **Urs Konig, Cowo Electronics, Munsterstrasse 4, CH - 6210 Sursee, Switzerland.**

EUROFAIR

The National Dutch QL Users' Club, Sin_QL_air, send new year greetings to friends of the QL everywhere, and announce their forthcoming workshop (23 February 1991) and Continental Microfair (6 April 1991).

The workshop will be held at the group's usual meeting place, Ir. A J Versfelt MTS (Technical School), Grebbeberglaan 15, Utrecht, Holland, from 10 am to 4 pm on Saturday 23rd February 1991. The group will be showing off its Sigma_QL68008 FN10 project, in which 20 members are working to build a QL model with up to 4Mb of memory, and such features as eight ports for external connection, and slots for three 1Mb memory cards including a 68070 by 16-bit datapad. The group's project manager is J B Janssen, Wilgenlaan 72, 4871VE Etten-Leur, Holland.

The group is also encouraging and testing the development of new keyboard interfaces and different types of keyboard.

The Continental Microfair will be held at St Joriscollege, Roostenlaan 296, Eindhoven, Holland, near the Antwerp-Eindhoven motorway (Leenderheide junction) and opposite the local zoo, on 6th April 1991. Flights direct from Manchester airport to Eindhoven airport have recently opened, the group notes.

The Microfair is purely a QL fair and user groups, developers and dealers are invited to display and explain their activities and projects. Reasonable costs and on-site catering are promised. For more information, please contact either J J van der Modengraaf, Mullerweg 17, 5624 JC Eindhoven, Holland, or C H M Biemans, Elzenstraat 5, 5461 CL Veghel, Holland.

NUMBERS FIXED

Author Simon Wallis has corrected some listing errors which appeared in 'Internal Numbers', *QL World* September 1990.

"I do not have a printer which works with my QL, so I copy-typed the listings onto another computer and printed from there," he says. "In doing so, I find I have introduced several errors, of which four are significant. I apologise for the delay - I have been out of the country for a while.

"With the following corrections, listing 5 runs correctly and outputs the 100th Fibonacci number in just under 30 seconds:"

Line 1665 has been omitted. It should read:

1665 READ data_item

Line 1760 should read:

1760 DATA 10241, 24832, 138, 21381, 6396, 0, 62988

Line 1810 should read:

1810 DATA -24, 28672, 20085, 14337, 24832, 72, 87974

Line 1850 should read:

1850 DATA 6144, 10242, 24832, 28, 21381, 30463, 93090

"Listing 6 will run correctly and achieve all of its stated objectives given the following changes to listing 5:

In line 1480, change num3base to num2base

In line 1510, change num3base to num2base

In line 1310, change num to num3

In the meantime, reader L Atkins of Biggleswade has independently come up with the same corrections to get the program working.

"While it seems unlikely that a calculator, capable of handling numbers to an accuracy of twenty, thirty, fifty or whatever significant figures, is unlikely to be of great practical use, it is a fascinating demonstration of the power of a small computer when programmed by an individual with sufficient imagination," writes Mr Atkins. "While I appreciate that a division function would be very difficult to arrange, it would make for completeness. Is it possible that Mr Wallace could be persuaded to attempt such an additional function at some future date?"

We'll have a word with him.

THORWAY

Further from Norway, we hear that Malcolm Smith is producing an occasional newsletter dedicated to the Thor. In fact, one has just arrived here. Titled *Mjolner - The Unofficial THOR User's Magazine*, the 8-page A5 booklet introduces the author, comments on certain problems with the Thor (he recommends a BBC printer cable for owners who cannot get a CST Thor cable), and asks for response and comments from Thor owners. The question of subscriptions will be held over until he hears from other owners.

Contact Malcolm at Statsrad Ihlensvei 66B, 2010 Strommen, Norway.

Mjolner, incidentally, is the hammer that the god Thor used to sort out his problems.

VALHALLA REVISITED

Simon Goodwin reports on the pros and cons of the 16 bit Thor system

After a dozen variations, the Thor XVI seems to have settled on a standard rom configuration. I have been using Argos 6.41 on my 1 megabyte Thor XVI since August 1989, and I'm used to it. As usual, I'm writing this in *Xchange Quill*, which comes with the machine.

The Thor XVI is a 'Super QL', first manufactured in 1987. It is based around a 68000 processor running at 8MHz, like the Atari ST, so it is typically about three times faster than the QL.

The display format matches the default QL screen, but no Sinclair parts are used; in fact the Thor uses the same lsi chip as Acorn's BBC Micro. There is an extra graphics MODE 12, with the same 256 square resolution as MODE 8, but 16 colours, courtesy of a PC-style rgbi monitor.

The Thor XVI was designed by David Oliver and Graham Priestley, at CST in Stevenage. After a year production abruptly moved to Bruel and Kjaer in Denmark, leaving Graham out of a job.

From the start the Thor's most ardent advocate has been Hellmuth Stuenkel, once main Sinclair agent in Denmark. His companies Dansoft and Thor International have sold hundreds of XVIs, mainly to business users in Denmark, Germany, Italy and the Soviet Union, where Hellmuth hopes to establish a 'screwdriver factory' assembling machines under licence.

Moscow firm YC Integral has reputedly offered to trade Soviet chips, power supplies, grinding wheels, pallets and potatoes for Thor components. Such 'barter trade' is commonplace - a rival business in Leningrad makes Spectrum clones in thousands for Soviet schools, bankrolled by sales of crystal germanium to Switzerland - but the Thor deals have been slow to crystallise.

Meanwhile the XVI has undergone 'production engineering' to suit Bruel and Kjaer's computerised production line. The renowned Danish instrument-makers pride themselves on flexible automation. Press one button, it makes a Thor... press another and you get a microphone, a Geiger counter or possibly salami on rye.

The re-design exercise pushed up the price of individual machines but meant that new two megabyte memory cards could be used in the three internal sim slots, boosting the upper limit of memory capacity from 2 megabytes to 6.5 megs on Danish models. A SCSI software upgrade allows any Thor XVI to control eight hard disks up to 100 megs each, rather than the

one or two 40 meg drives that were the limit for early Thors.

The current 16 bit Thor XVI model is planned to be superseded by the Thor XX, a 68020 machine with four times the graphics resolution of a QL; I saw the specification a year ago, but have yet to see the product. This is the fourth 'fully designed' 32 bit QL clone I haven't yet seen. I hope they're all going to be compatible!

Whatever the outcome of Hellmuth's plans, many QL enthusiasts have upgraded to the Thor XVI. The main alternative is the Atari ST with *Futura* QL emulator, based on JS Qdos. The Thor boasts QL network and expansion connectors, and a port for up to eight SCSI devices.

The Thor system rom has been stable for almost two years, so I have had time to get used to the peculiarities of Argos 6.41. This article charts my practical experience of Argos, much like my tours of *Minerva*, another close analogue of Sinclair's QL system. In fact I network them together, with *Minerva* serving the Thor, providing access to six drives, QLd1, to QLd6, and any other devices I care to open over the net.

Like *Minerva*, Argos is inherently faster and more reliable than Sinclair's Qdos. Despite contrary protestations, both owe much to Sinclair MG roms, released late in 1985. They proclaim compatibility, while adding features to existing Basic commands.

Most 'corrections' or changes bring deliberate advantages and accidental disadvantages. Rom-revisors have a tendency to be alternately reticent and self-congratulatory, but forewarned is forearmed. New features are great, as long as you know their good and bad points.

There's no point in feeling miserable about your AH rom if you wouldn't use, or notice, the new bits or don't fancy dodging new pitfalls. Compared with version FB, Qdos 1.00, they're all almost right; you are doing fine if your machine runs all the programs you like to use.

If I hadn't got my Thor, I'd have to buy another computer. That is saying a lot, as my desk layout owes a little to Rick Wakeman; I can reach six keyboards at the moment, and I'm typing on the Thor, which has a PC-style keyboard on a curly cable.

The Thor has no cartridge port, but there is an internal 8-bit expansion rom socket. Mine contains a 64K *Intrrom* containing Talent's and my text editor, *Speedscreen* 6.24, *Turbo Toolkit* and parts of *Toolkit 2*.

The *Intrrom* came with the machine, but it is no longer very useful: the *Toolkit* code is out of date and gets copied to ram in order to run, while *Speedscreen* is sluggish because it runs from 8-bit Rom.

CST advertised *Intrrom* to XVI owners, and it fits the socket, but it is intended for the Thor 8, and not really compatible with the latest roms. If I initialise either of the *Toolkits* after any program stops with an error in expression, the machine crashes and must be reset.

Turbo Toolkit 1.42 was pretty good for its time (1986) but it does not communicate with versions of the *Turbo* compiler since version 2.01. The latest version 2.04 uses the Argos Thing list, which starts at PEEK_L (SYS_VARS+352), for communication between PARSER, CODEGEN, CHARGE and EXECUTE, and toolkit and compiler must match before *Turbo* can work.

I have had no direct dealings with Digital Precision since 1987, but do hear from Chas Dillon, who maintains the *Turbo* parser. The last version of *Turbo Toolkit* I have seen is Runtime 3.20. It is romable but has extra baggage in the form of micro-drive access routines and other things Chas wants in everyone's computer. I re-coded DEFAULT_DEVICE to alias PROG_USE, so you should now be able to keep *Turbo* in a sub-directory.

The *SuperToolkit* extensions and defaults called up from the Thor's *Intrrom* by TK2 are convenient, if you're used to them on the QL, but many come as standard on the Thor. Argos 6.41 includes *Toolkit 2* homonyms like LRESPR, CLOCK, SAVE_0, OPEN_OVER, OPEN_DIR and CLOSE without parameters. ALTKEY is not provided, and QJump's code will not run. However a tiny key-defining utility called FNKEY20Z is buried in the Quanta library, and that does work on the Thor.

Qram does not suit the Thor XVI, and locks the machine if you try to load it. The Thor family has its own windowing system, so that tasks do not necessarily over-write one another's windows; this uses an extra 16 bytes of data at the start of each window channel definition. EXTOP conceals the extras from add-on code, so DIY *Toolkit* functions work just as on a QL.

TurboQuill Plus is not happy on the Thor, although it prefers the Thor character set. This is little loss as *Xchange* has its own 'glossary' feature, and runs noticeably faster than the original *Quill*.

QPac-2 does not work properly on the Thor XVI, but it is apparently happy on rare Thor 20 or Thor 21 systems. I think this could be fixed, if there was a demand;

even on the XVI I understand that 'button functions' allow access to QPac utilities, and 'hot keys' can be accessed by tapping SYS REQ to swap to and from the controlling task, before pressing the required combination.

Thor windowing can be disabled with POKE SYS_VARS+133,-1. Subsequently opened windows are not saved when tasks swap, and tasks continue to run even if their windows are obscured. I put this POKE in my BOOT program, so that Basic windows are protected but other tasks share the screen, QL style.

All the tasks I use continuously can easily be provoked into 're-drawing' their displays fast on the Thor; Devpac and Qlipboard respond to F4, like most text editors, while Psion programs expect F10 (Shift F5 on the QL). Overlapping windows allow continuous multi-tasking, and save hundreds of kilobytes that might otherwise be clogged by stored displays.

If free memory is really at a premium you can close and re-open the Thor Basic windows, to release their screen-stores. Put:

```
CLOSE #2;CLOSE #1;CLOSE #0:
```

on a single program line, before the POKE, and

```
:OPEN #0,CON:OPEN #1,CON:OPEN #2,CON:WMON 4
```

thereafter. Make sure all eight statements

are on one line. Don't try this on the QL because Sinclair did not expect those channels to close; it won't help, anyway, in the absence of the windowing system.

The first Thor XVI rom I encountered was version 6.30, which had many similarities with MG Qdos. It came in two 64K chips, with lots of empty space. Later roms show signs of large-scale re-coding in the Basic interpreter, but still leave around 40K of rom space fallow.

I was fairly happy with the original 6.30, although it would not run QMon; I guess it makes assumptions about the processor which suit the 68008, not the 68000, as it works at a very low level.

Investigations of the Thor Rom are handicapped by the fact that I still can't get QMon 2 to run on the XVI. The first versions were essentially MG Roms with altered device drivers and modified tables. Earlier Thors used Sinclair JS Roms.

Slowly Argos has taken on its own character, like Minerva. It works much as before – although intermediate versions contain howlers – but system calls are faster and often more reliable, or more powerful. The problem, as with all upgrades on an established standard, is that programs need to run on old QLs and Thors as well as XVIs if they are to be commercially successful. This means that new features are rarely used in published software.

Small rom system improvements help to make the Thor seem even faster than benchmarks against the QL suggest, but in general the Thor will not run notorious

programs that expect a 68008, a 'key' cartridge, or particular Sinclair roms.

The vast majority of QL applications run perfectly on the Thor, and I'm most surprised if I manage to crash the machine; even if I do something daft Argos usually spots an 'exception' and suspends the task. Of course it is possible to crash it if you know how, so I've cited the methods I have found by accident, and said how I learnt to live with them.

There are some potentially fatal bugs in the current floppy disk driver. Direct sector access to DIM'd strings can crash the driver, as I noted in December; the same problem occurs if you try to load a sector into a LOCAL string with a preset maximum length, yet the target must match the sector size. The answer is to preset the length with FILL\$, rather than DIM, as in FAST_FORMAT_BAS, or use Turbo's INPUT\$ function. Compiled tasks seem to read sectors OK, thankfully.

Disk changes can confuse the Thor, and may even make it crash. If you accidentally ask for the directory of a disk that has not been formatted Argos reports an error, but something gets into a muddle, because the machine crashes if this happens repeatedly. Similar problems may crop up if you swap disks without closing files. It took me quite a while to reproduce this fault by swapping disks and interrupting commands; you should be fine as long as you heed the initial warning.

My machine has two 720K floppy drives, and I'm saving up for a hard disk; Chas

Thor Xchange displays part of the Russian Thor brochure in a mixture of cyrillic and Latin script.

HELP press F1	CURSOR move ↕	TEXT Insert: Type at █	TYPEFACE press F4	COMMANDS press F3
PROMPTS press F2	with ↑↓ keys ↕	New para: Press ENTER	GLOSSARY press F5	XCHANGE press F6
		Delete: CTRL & ↕		
		Change mode: F9		

ЗВМ "THOR XVI" обеспечена языком программирования "ThorBasic" (13), высоко-структурным процедурным интерпретативным языком программирования. Язык подлежит расширению и обеспечивает совместимость снизу вверх с языком программирования "SuperBASIC" (4) и компилируется с компилятором "Digital Precision's Turbo compiler" (7).

Язык программирования "ThorBasic" естественный язык коммуникации с операционной системой "ARGOS". Комплект команд, содержащий одну отдельную строку или тысячи строк, можно выполнять с встроенным интерпретатором или компилировать с помощью компилятором "TURBO Compiler".

----- end of page 3 -----
Для показа относительной скорости ЗВМ "THOR XVI", мы включаем эталонное тестирование журнала "PCW" (8). Цифры представляют затраченное время (в секундах) на выполнение алгоритмов стандартных эталонных тестов; поэтому чем ниже цифры, тем лучше:

Эталонные тесты

MODE: INSERT WORDS: 3390 LINE: 0 PAGE: 4 TASK: QLWorld

Dillon has used hard drives from the start, and found the early version rather limited; it lacked direct sector access, and would only work with certain Rodime SCSI drives.

Version 6.37 introduced better SCSI, supporting more than two hard drives with a faster 1:2 sector interleave. This pushed the top speed past 200K per second, which is good news when you could have 800 megabytes on-line. Even then it would take about an hour to read the lot, flat out.

Chas enthusiastically re-formatted his 40 megabyte disk and discovered that 6.37 and 6.38 had Name Table inconsistencies that stopped them working with Turbo. After some frustrating phone calls 6.39 became the 'definitive' Thor rom, for Chas and some others.

Thereafter Thor International embarked upon a major re-write. The short-lived Argos 6.4.0 was almost un-usable, with 225 bugs reported, but 6.4.1 seems stable.

Argos 6.4.0 had a terrible bug which could easily crash the Thor Basic interpreter. FILL\$ was always OK under Turbo which has its own FILL\$ code, but the bug affects QLib, which calls the rom.

I am glad to report that multiplication by zero works in Argos 6.4.1, whereas 6.4.0 could get weird answers due to a errant optimisation. This system bug should have upset Turbo tasks, as they call the affected routine, RI.MULT, but in practice Turbo already traps the special case and keeps it away from the rom. In theory the bug could affect graphics, as they use RI.MULT internally, but in practice you should be OK as the scaling code should never try to multiply by zero.

6.4.0 could only manage Turtle graphics in the default window #1, but 6.4.1 works as happily as any QL, and faster. Thor graphics still seem based on the Tortoise routines GST supplied to Sinclair in 1983; Argos has the CIRCLE bug which is in *Lightning* and Sinclair roms, and recently fixed by Minerva.

Another bug in 6.4.0 used to crash the Thor if the '!' separator was used by INPUT or PRINT to a non-console channel. I guess the Danes made a mess of picking up CH.WIDTH. Predictably, Turbo and Supercharge have their own way of doing this, so compiled programs are not affected. Thor Basic 6.4.1 is OK.

Argos 6.4.1 has been with us for over a year, and now that the slim resources of Thor International are aimed at 32 bit horizons an imminent upgrade seems unlikely. This makes it important to know the bugs in the latest version, code named CS - the third character is a space.

I am not sure what the C stands for, but would hazard a guess that S is for 'Stuven'. The first Thor XVI ROM was version PT, presumably in honour of David Oliver's secretary and fiancée Penelope Tsatsaris. Early versions of 6.4.1 were version PO; as far as I can tell messages are changed, but not code.

Current Thor XVI systems require version 1.07 of the 32K eeprom that controls the 6802 second processor. This handles the SCSI, network and serial ports, com-

municating with the main 68000 processor through 8K of static ram at 68000 address 16744448. The shared ram is faster and more reliable than the serial protocol that links the QL's 68008 to the 8049 co-processor.

Interesting things are passed in the shared ram; POKE -30976,VOL sets the dac volume level used by BEEP, ranging from silent if VOL=128, via quiet at 129 and 130, to a car-alarm level that shakes the metal case, set with POKE -30976,255. IO rom 1.07 has been current since the device driver upgrade of Argo 6.37.

I got my set of eeproms from Thor International in 1989, but have not been able to contact them recently to check availability. You should note that Danish and UK Thors have different eeprom pin assignments. The QL version used a QEP-3 feature to scramble the priorities of signals, simplifying the board layout, but the production engineers swapped address and data lines back to conventional assignments.

Thor International are proud of the Bootmenu program that they supply as a 'front end' to configure the system, but you don't have to use it. It does look nice in Russian, though.

At first the Thor XVI used to crash horribly if Scroll Lock was pressed before the BOOT file was found. Version 6.4.1 just gets its windows in a muddle, ending up with #0, #1 and #2 all full-sized and overlaid if you press Scroll Lock while the roms are initialising. WMON 4 sorts that out. The start-up time to initialise 1032K ram and 188K rom is about 14.5 seconds.

Another interesting bug cropped up when I loaded *Taskforce* on the Thor. The program stopped with an error because of the Basic definition of the function SYS_VARS. This over-rides the rom resident function of the same name, so the command disappears from the Name Table if NEW or LOAD is entered thereafter.

Subsequent Resident procedures and functions in the Name List are re-numbered. This 'feature' means that you can delete resident procedure and functions, by defining them as Basic, then deleting them. That can be useful, as long as you don't do it by accident!

There is something wrong with the network File Server in the Thor rom; it works perfectly with the QL serving, but not the other way around. After a while the Thor falls over. I have not been able to try this with two XVIs; it may signal incompatibility between the Thor and my version of the QL SuperToolkit rom, at the other end of the net link.

Be careful not to read big tasks or other binary files into code that expects lines of data, like INPUT or VIEW. Thor Basic goes haywire unless it gets a new-line character before 32K is up. It's easy to end up with negative-length strings and other Russian roulette situations, unless you make sure you get an end-of-line character every 32767 bytes.

A similar bug can affect machine code directly, as the SERIO routines get in a mess if asked to perform IO.FLINE for

more than 32K. This can cause problems with devices like SER, PAR, MEM and PIPEs. The same bug is true of QL roms - the fix would slow down quite a few devices, so it's probably not worth it. The new DIY Toolkit Qlipboard task avoids this problem.

The Thor has WHENERROR code similar to that in MG roms, but it does not seem to suffer from the serious bug that causes an infinite loop in the SuperBasic interpreter when you try to trap an error inside an expression, on a Sinclair rom.

The Thor's implementation of 'WHEN Variable' is similar to that in MG roms. The condition after the WHEN is evaluated whenever an assignment changes the initial variable. WHEN X>5 runs successive statements up to the END WHEN whenever X gets set to a value greater than 5.

There are some limitations. INPUT, READ and other implicit parameter assignments do not activate WHEN. The system fails if 20 or more variables are monitored, like JS and MG roms. It works OK with 19, but 6.4.1 croaks *Exception 0* at 12F2 when it calls the twentieth block, or attempts to set up a twenty first.

The problem is the set-up code which allocates data for WHEN blocks. QView ace Laurence Reeves explains that space is allocated in lumps from the variable values area; each lump holds the details of up to 20 WHEN blocks, and Sinclair failed to test allocation of the second block.

There's a subtle difference in the way Thor Basic handles PRINT TO, compared with Sinclair systems.

PRINT "x"; TO 1;"y"

gives:

x y

(with a space between the letters) on a QL in SuperBasic, whereas the revised Thor code prints:

xy

The bug stems from the Danes' ambitious re-write of the awful routine shared by PRINT and INPUT. Thor International had to re-write that code as it's a big, ugly, characteristic monolith. I re-wrote it for *Supercharge* because I couldn't understand it, and it wouldn't do everything I wanted. I ended up with 22 optional chunks of code in the library of routines the compiler uses to re-express PRINT and INPUT. Tony Tebbey says the Sinclair code started life as a debugging routine, intended for checking the Variable Values area.

The problem is that the TO separator always sends at least one space on the QL, as noted on page 58 of Jan Jones' Def Guide. Thor International's code sends nothing at all if the print position already matches the 'TO' parameter. That's quite logical, but not compatible.

This is not a big bug; Turbo and Supercharge tasks refuse to call either Sinclair's

or Dansoft's code for PRINT and INPUT, so compiled tasks work consistently on QL and Thor. However, QLib tasks show the fault, as they always use resident procedure code.

The Thor's new code seems wrong to me, because I find it convenient that TO should separate columns. If the parameter is no more than the current position, one column has probably overflowed, but I still don't want it to run into the next one. I may be biased because Turbo works that way. It's a minor quirk, as long as we all know about it.

The Thor does offer one improvement over QL PRINT and INPUT. It lets you put new channel numbers part-way through a statement, so:

```
PRINT "Your Name ";: INPUT #0,name$ :  
PRINT "is ";name$
```

becomes:

```
INPUT "Your Name "; #0,name$ \ #1: is  
";(name$)
```

The Turbo library can handle this already, if the parser grammar is tweaked to recognise the new syntax. The library forms the majority of CODEGEN TASK, and hasn't changed since 1987. Despite later parser improvements, it still holds untapped library routines and optimisations.

I have dispatched details of Turbo grammar changes that should allow the Thor's extended syntax for CLOSE, COPY, INPUT, PAUSE, PRINT, SDATE and WINDOW. A few other changes will allow similar improvements when programs are compiled on Minerva, but the resultant code may not work on other roms, as it might rely on the modified expectations of the resident procedure or function.

Even Supercharge runtimes support multiple channels in a single PRINT or INPUT, but some other parameter variations will have to be passed on to the rom for parsing at run-time, like *DIY Toolkit* extensions. Whenever possible, Turbo checks the parameters of rom and Turbo Toolkit commands while parsing, unlike Q-Liberator or the interpreter.

I believe that the true test of a compiler is how it copes with incorrect programs, so Turbo aims to report potential problems whenever possible, rather than check for them repeatedly, ever after. Thus current versions of Turbo spot and reject parameter sequences that would upset Sinclair keywords.

Before the compiler re-compiles itself a separate program reads a textual description of know commands, their separators and parameters. This utility generates DATA statements which are MERGED into the parser before compilation. The internal 'grammar' allows better checking, saves writing lots of ad hoc code, conserves memory and makes changes easy, if DP considers them worthwhile. Of course, QLib 'compiles' the new variants without murmur.

The Thor XVI has real serial and parallel ports, much better than SER1 and SER2

on the QL. The protocol is now handled in hardware, as on all but the cheapest modern computers, and the SER device name has been extended with many new options, like SER2exb75b1200cf 32K!

The serial ports handle 5 to 8 bits per character, with optional parity generation and checking, XON-XOFF with or without handshaking, separate input and output baud rates, optional translates and buffering.

The Thor has an extended TRANSLATE facility to sort out printer and modem codes. You can specify a prologue or epilogue message up to 32K long, which will be sent when the channel is opened or closed.

Thors are designed for multi-national appeal, and the extensions SET LANGUAGE\$ build on TRA, offering rom messages and display characters in 12 variations, including International, British, French, Spanish, German, Greek and Russian. The collating sequence used by INSTR and case-comparisons is still wrong for many Greek and Russian letters, but nowhere near as wrong as it is in British QL roms.

The file transfer commands WCOPY and COPY have been rewritten to make better use of buffer memory, and they can now combine files as well as copy them.

Some Thor improvements are reminiscent of Minerva. PAUSE now allows a channel parameter, while SDATE TO accepts a single parameter date, in seconds since 1961.

Some long-standing bugs fixed in Minerva still occur in 6.41. The Thor crashes if you allocate a negative amount of space with ALCHP, use more than 20 WHEN clauses, or try to PRINT "O" and 1/O. Graphics routines have the same bugs as Lightning and early QL roms: try CIRCLE 80,80,80,6,1 if you want a laugh!

The XVI emulates KEYROW and IPC calls to the second processor, although BEEP pitches rarely correspond. Thor-specific programs are encouraged to read keyboard data directly - this is much faster and more reliable than MT-IPCOM, which is provided for compatibility with QL programs.

Useful functions SYS VARS and NET ID reveal the address of the system variables and the net station number. In practice you can find the later with the former, on QL or Thor: NET ID reads SV.NETNR, like PEEK(SYS BASE+55). 6.41 fixes one old inaccuracy, which I would hesitate to call a bug; at last COS(P1/2)=0, previously COS(P1/2)=40. Who cares?

SDATE now sets the battery backed clock; the old Thor command SET_CLOCK is still there, but you don't need to use it. NO_CLOCK gets rid of the multi-tasking CLOCK job.

The new function IO_TRAP is a passport to Argos - a variant of the CALL command which lets you pass register values to and from any device, via TRAP #3. I have given examples in recent Toolkit conversion notes. This function is much-needed as undocumented parameters of CLS, PAN and SCROLL are now rejected, whereas they do useful things with cursors and file

pointers on Sinclair systems. You can do all these tricks, and more, with 10_TRAP, but you have to change your program to suit the Thor.

The Thor CHR\$ function only accepts parameters between 0 and 255. This correction stops CHARSET BAS, on the Quanta library disk UTIL GEN1, as it tries to print character codes 256 to 259, which the QL happily prints as splodges, and some Minervae render as NUL to F5. Strictly this is a fix for a QL bug which a few programs exploit, but it could be the cause of otherwise inexplicable range errors.

Similarly the Thor's POKE rejects parameters beyond the range -128 to 255, allowing signed or unsigned byte values; the QL POKES the least significant byte of any integer supplied. Argos 6.41 limits POKE W to values from -32768 to 65535, while QLs, Turbo and Supercharge tasks accept any long integer and store the low word.

The revised CSIZE only tests the least significant bit of the vertical size parameter, so CSIZE 2 - 32768 has the same effect as CSIZE 2.0. Odd integers select double-height text - you don't get any new sizes.

The Thor's WINDOW allows border width and colour as two optional extra parameters. TOP WINDOW tells the Thor screen shuffler which window should take precedence; peculiar 'windows', these.

The extra colours in MODE 12 are accessed by passing fractional colours to the Basic commands INK, PAPER and STRIP. INK 7 sets grey ink, and INK 7.5 is white. PAPER 7.25 and PAPER 7.75 select alternate grey and white stipples.

This convention confounds Turbo, which tries to use fast integer arithmetic as much as possible. Turbo minimises interpretation and parameter passing by converting sets of stipple values into composite integer colours, using its own code. This is great for QL MODEs, but means that the fractional part is ignored in MODE 12.

Even the Thor uses integers to represent colours internally. Halves and quarters are moved to bits 14 and 15 of D1 before calls to SD.INK or SD.PAPER, so you can use all the MODE 12 colours in compiled programs, as long as you select them with calls to 10_TRAP. This is little hardship as MODE 12 programs are inevitable Thor-specific.

This trial confirms that the biggest challenge for would-be Rom improvers is the pressing need for compatibility with existing programs. There are two 'best' Rom sets for the Thor XVI: 6.39 and 6.41. If you have 6.40 you need an upgrade; if you use Thor Basic or QLib you probably already know that! Both my recommendations have some annoying bugs, but they work better than intermediate versions of Argos, and include the vital SCSI improvements.

I hope to have a hard disk running on my Thor before the end of the year, and plan to report on the upgrade in a future article. In the meantime I am keen to learn the experience of other Thor owners. Please write, care of *QL World*.

As I indicated in an earlier column, if you have *SuperToolkit II* and a bit of extra memory the ALTKEY function can be used to save a lot of repetitive key pressing for routine operations while using the Psion programs – it's particularly useful with Quill. 'Strings' of characters can be assigned to particular <ALT & key> combinations during your boot and these remain available until they are redefined or the machine is switched off. Some 'Front End' programs like *Taskmaster* or *Qram* use similar combinations for other purposes, so you may have to change some of my suggestions, but there are plenty of others available since pretty well any key combination that prints a character can be used, this includes the CTRL and CTRL & SHIFT combinations that generate the funny foreign characters – although you may need a third hand to manage some of the combinations and ALT at the same time. (Remember, you have to be holding down the 'modifier' keys *before* you press the main letter key.) One pair you must avoid is <CTRL & 7>, since adding ALT to this combination freezes the machine completely, and a complete reset is required – even Minerva's soft reset is frozen.

The best way to assign your keys is to set up a special SuperBasic procedure that is called in the boot. *Listing one* shows a skeletal form for such a procedure, you should be able to flesh it out to suit your own needs. Some of the keys you have to press in Quill do not correspond to printable characters, but this does not mean they do not generate codes like the letter keys, it's just that they are intercepted as control codes by the operating system and never reach the screen driver. ALTKEY inserts codes into the keyboard buffer just as though they have been typed in, so you can still use CHR\$() to put these codes into an ALTKEY

definition, and, even better, you can assign them to strings just like printable characters. The only keys that do not generate codes in this way are the 'modifier' keys SHIFT, CTRL & ALT which are used in combination with other keys and modify their values.

CAPS LOCK is also treated differently, although it does generate a code (225). I like to assign the special codes for the function and arrow keys to short string variables with meaningful names at the start of the procedure. This avoids having to use the relatively meaningless CHR\$() form during the body of the ALTKEY assignments; I find F3\$ much easier to remember than CHR\$(240) for example. (In case you didn't see my earlier article, you can find the code generated by one of the "special" keys or combinations using a mini basic program that you can type in directly as a command.

```
rep k: print code(inkey$(-1)),
```

This will print the code corresponding to any key or combination (even <ALT & key>) until you press <CTRL & space> or look them up in the manual.) Most of them can be incorporated into your string with full effect, and this is the power of the method since even the non printing keys like <F3>, <up arrow> etc can be included in your assignment. If the ALTKEY assignment includes more than one string separated by commas an <ENTER> (=CHR\$(10), also known as "line feed") is inserted between each string, and you can use this instead of lf\$ if you like (see line 150). If you analyse

Using Altkeys in Quill

the lines in the Listing you will see that the string can be normal characters in quotes, eg 'address', a string variable eg F3\$, or a function returning a string – FILL\$() is the most useful here since it enables mimicing of repetitive keystrokes (lines 165 and 185). These can be linked together with &s to make up a longer string.

To create an ALTKEY definition you should load Quill, work out how to do what you want to do from the keyboard, and then write down the exact sequence of key strokes required (including <ENTER>s). You then have to construct a string that duplicates them exactly. I will go through the effect of those in the listing line by line but leave the reader to analyse them to work out how they work. Since there is often confusion in listing between the numeral 1 and the letter l I should point out that they are all letter ls, except where obviously part of a number.

110 – 130 Assign values to string variables for use later in the PROCEDURE

Settings

135 <ALT and a> loads a file called 'address_doc' from the default device. To my mind this is the answer to those who, like Reg Gilbert of Dartmouth, Nova Scotia, write in asking for a way to modify Quill to change the default margin and justification settings. Whatever you do there will be only one default set available. But you can save as many blank files set up for different purpose as you like, and it takes only a moment to load them if you set

them on different ALTKEYs. In case 'address_doc' contains my standard letterhead.

140 <ALT and b> Selects/deselects the bold typeface; the other typefaces and 'paint' can be set up similarly. You may not think this worthwhile since all it does is to replace two consecutive keystrokes with two simultaneous ones, but I find it avoids confusion between F3 and F4.

145 and 195 <ALT & B>, <ALT & 2> etc. Move the cursor to the bottom of the text, or a specific page number.

150 <ALT & d> goes straight to the "Delete" option in 'Files' and lists the files on the default device on the screen.

155 <ALT and E> threatens to delete from the cursor to the end of the paragraph. It requires one more <ENTER> to complete this irrevocable operation; press <ESC> if you have got there in error! One other problem is that after using Erase (or Copy) the program may freeze on repeated use of <up arrow>. Save and re-load to avoid this – this is made easy by the next macros (slightly out of order):

175 <ALT and s> inserts a Yen sign at the cursor position and then saves the current document under the name 'current_doc' on your default device. It then uses the search routine defined at line 130 to find the Yen sign, delete it and put the cursor back exactly where it was when you started. It assumes that there is already a file called current_doc and overwrites it; if there isn't then it will also add a y right at the beginning of your document. Remember to save under a distinctive name before you log off. If you are an international

PSION SOLUTIONS

Howard Clase uses SuperToolkit 2 to save time in Quill.

financier and use of the Yen sign proves an embarrassment then it can be replaced with any character (or characters) of your choice.

160 <ALT and £> re-loads the file called 'current_doc' and then searches out the Yen sign as above. I chose the £ sign for this since it is out of the way and difficult to strike by accident - the active file in Quill is overwritten by this one!

165 <ALT and m> moves the left margin five places to the right and the right margin ten places to the left. You could use this idea to re-set the default margins, justification etc., but it is only relative to the current settings and I think that

the solution under <ALT and a> above is better.

170 <ALT and S> prints the current document whole to printer.

180 <ALT and S> like <ALT & s> but doesn't change the name. NB the Yen sign will be present in the saved file.

185 <ALT and Y> prints 'Yours sincerely' and your name in the appropriate places, leaving 6 lines for a signature.

190 <ALT and ?> lists the files on the default device on the screen and waits for a file name to load. Press <ESC> to get out of it.

If you have worked through all these then you should now be able to design a set to suit

your own needs.

Incidentally these macros are much quicker if you have pressed <F2> to remove the instruction panel from the top

of the screen, otherwise you have to wait while it needlessly prints out all its messages. To print the Yen sign press <CTRL & SHIFT & . (full stop)>.

```

100 REMark ~~~~~
105 DEFine PROCedure ~~~~~ Altkkeys
110 f2$=CHR$(236): f3$=CHR$(240): f4$=CHR$(244)
115 la$=CHR$(192): ra$=CHR$(200): REMark arrows
120 up$=CHR$(208): shdn$=CHR$(220): del$=CHR$(202)
125 lf$=CHR$(10): tab$=CHR$(9)
130 search$=f3$&f3$&'s'&'&lf$&lf$&del$
135 ALTKEY'a',f3$&'l'address'&lf$
140 ALTKEY'b',f4$&'b'
145 ALTKEY'B',f3$&'gb'
150 ALTKEY'd',f3$&f3$&'fd?',''
155 ALTKEY'E',f3$&'e'&lf$&shdn$&lf$
160 ALTKEY'f',f3$&'l'current'&lf$&search$
165 ALTKEY'm',f3$&'m'&FILL$(ra$,5)&'r'
    &FILL$(la$,10)&lf$
170 ALTKEY'P',f3$&'p'&lf$&lf$&lf$
175 ALTKEY's',f3$&'s'&f3$&'sflp1_current_doc',',',',yy'
    &search$
180 ALTKEY'S',f3$&'s',',',',yy'&search$
185 ALTKEY'Y',lf$&FILL$(tab$,2)&'Yours sincerely,'
    &FILL$(lf$,6)&FILL$(tab$,4)&'Your Name'
190 ALTKEY'?',f3$&'?'&lf$
195 ALTKEY'2',f3$&'g2'&lf$: ALTKEY'3',f3$&'g3'&lf$
200 END DEFine
  
```

QL SUPERTOOL KIT II by Tony Tebby

Over 118 Commands:- Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network, 16k Eprom Cartridge Version @£24.15d
Configurable Version on Microdrive @£24.15d

MIRACLE SYSTEMS PRODUCTS

OK Disc Interface (Inc Tool Kit II) @£99.82b
QL Centronics Printer Interface @£28.75d
QL Expanderam 512K ThruCard @£99.99e

QL HARDWARE

Single 3.5" Disc Drive & (Own PSU) @£103.50a
Dual 3.5" Disc Drive & (Own PSU) @£188.60a
3.5" DS/DD Discs - 10 off @£9.20c
Q POWER REG. The only real solution to you QL overheating (switched mode power supply run cold) £24.15c
QL Keyboard Membrane @£11.50d
QEP III Advanced Eprom Programmer @£121.90d
Care Eprom Cartridges each @£5.75c
ULA Chip ZX6301 @£15.64c

PRINTERS

STAR LC 10 Mono @£179.40a
STAR LC 10 Colour @£224.25a
CITIZEN SWIFT 24-pin Colour @£356.50a

SOFTWARE 87 (State MDV or Disc)

TEXT87V.300 (Requires Min 256k) @£60.00d
FOUNTEXT 89 @£15.00c
FOUNTEXT 88 @£25.00c
TEXT 87/FOUNTEXT 89/FOUNTEXT 88 @£94.99b
248B PRINTER DRIVER @£15.00c
Upgrade to Text 87 V.3.00. Return old copy together with @£25.00d

MONITORS (Price including lead)

Philips BM7522 Amber Hi-Res @£97.75a
Philips CM8833 Colour Med-Res @£253.00a
Philips AV7300 TV/tuner for above @£69.00b
Philips BM7502 Green Hi-Res £97.76a

HOW TO ORDER:

ALL PRICES INCLUDE VAT

By Post. Enclose your Cheque/PO made payable to CARE Electronics or use ACCESS/VISA. Allow 7 days for delivery

TONY TEBBY SOFTWARE (QJUMP)

QLFP (Micro/P disk interface upgrade) @£14.95d
QMON II @£23.00d
QTYPE Type/Spell Checker @£29.09d
QPAC 1-Desk top accessories, calander, alarm, calculator, typewriter, digital clock sysmon @£21.85d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITHE copies M/DRIVE - M/DRIVE @£14.95c
4MATTER + LOCKSMITHE copies M/DRIVE - DISC @£23.00c

The above programs are not for use in the UK

SIDEWINDER - High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text insertion @£19.95c
SIDEWINDER PLUS - (for expanded QL's) Includes all the features of above, PLUS multiple label printing, desktop publishing files and printer driver for 24" pin plus LC10 and Jx80 colour printers. (Please state 3.5" disc or M/D) @£23.00c
Upgrade to Sidewinder Plus @£8.05c

HEAT TRANSFER RIBBONS & PENS

Print your own T-shirt Design. Just print on paper and iron onto your T-shirt.
Star LC10NX1000, Rainbow @£17.25c
Star LC10NX1000, Black @£11.50c
Epson FX80, MX80, LX80, FX100/Okidata ML80 Citizen 120D/Star NX10, Black, Panasonic 1080/81 @£11.50c
Small 5 colour pen set @£12.65c
Jumbo 5 Colour Pen Set @£16.10c

READYMADE LEADS

RGB QL to Phono @£5.75c
RGB 8-6 pin DIN @£7.13c
RGB 8-7 pin DIN (Hitachi) @£7.13c
RGB 8-7 pin DIN (Ferguson) @£7.13c
RGB 8 pin to SCART (Euro) @£11.50c
6-way PCC 25-way 'D' (Printer-Ser 1) @£9.89c

QPAC II

QPAC II Main menu windows adjust automatically to size. Files, directory, view, print, delete, backup, jobs, pick, Rjob, sort, channels, things exec, wake, buttons, Hotkey, Hotjobs. Fully multitasking, allows many programs to run at once. Requires min of 256k expanded memory @£49.91b
Upgrade QRAM to QPAC II return master @£29.90b

AS REVIEWED QL WORLD
AUGUST ISSUE

CARE
ELECTRONICS
OPEN
9am-5pm Mon-Thu
9am-4pm Fri

800 ST ALBANS ROAD,
GARSTON, WATFORD,
HERTS, WD2 6NL.
Tel: 0923-672102
Fax: 0923-662304

Please add carriage
a=£11.50 b=£3.45
c=£1.38 d=£2.30



THE NEW USER GUIDE

To an outsider, the QL has a surprisingly large and loyal set of admirers. Those associated with the computer more closely are less surprised by its continuing appeal. Even with the prices of comparable computers tumbling, the QL still represents remarkable value for money, and even though the competence of personal computers has improved immeasurably since the QL's launch in 1984, it can still claim to be superior to industry standard machines in several important respects.

The QL's construction and the ethos behind the decisions to equip it with a cheap keyboard and low-capacity microdrives are now dated, but its operating system is still superior to those supporting many pcs. Its programming language is a delight compared with the vast majority of Basic dialects available today.

Many QL users take to the computer like ducks to the village stream, but a great many other enthusiastic QL supporters have not found it so easy to come to grips with their computer's facilities. They make full use of the Psion programs, they purchase commercial programs, but there is the constant feeling that this is a second-hand way of using a computer. Their goal is to communicate directly with the QL, to have it obey directly their own commands.

The major hurdle for most unfulfilled computer users is often the 'user guides', which build up expectations and then fail

to deliver. The *QL User Guide* is by no means the worst in this respect, especially in its most recent reprint, but nonetheless it is disappointingly incomplete in parts and, on occasion, downright inaccurate.

The QL User Guide was written by Roy Atherton in a concise and academic style which befitted the QL's image as a serious home computer. His task was presumably made more difficult by having to write against the clock and revise his work to accommodate the frequent revisions of language and operating system specifications which occurred before the QL's premature launch date. Certainly, his book on programming the QL is much better organised and easier to follow, although much the same in style as the User Guide.

Here at *Sinclair QL World* we have been aware for a long time, and are continually reminded by letters from readers, that a new User Guide would be extremely welcome. The aims of the new guide would not only be to correct the misleading parts of the original User Guide but also to present the subject from another perspective. If you failed to grasp Roy Atherton's insights you may be more enlightened by a different angle.

The prospects of any new book on the QL being a commercial success appear to be slight: there is no longer a large enough market to sustain a print run of, say, 25,000 copies which would be necessary if the book was to be available at a reasonable

price. A low-cost manual printed on A4 sheets and distributed by mail order looked a more promising prospect, but print and advertising costs made the venture a risk. Consequently, it has been decided to include the New User Guide as part of Sinclair QL World, to reach the largest number of QL readers at the most economical cost.

The New User Guide starts with a Beginner's Guide covering very much the same ground as that of the original User Guide. This will allow readers to cross-refer between the two to get the best of both approaches to the subject matter. Additionally, it allows us to point out in detail any corrections which need to be made to the original guide.

Computing is becoming a crucial skill in modern society, but it is one which does not come naturally to the majority of people.

The situation is analogous to driving skills — only the most talented become racing drivers, but almost every adult has the capacity to drive a car competently. Computing should be the same; do not conclude that just because the extraordinary achievements of programmers like Simon Goodwin, Johnathon Oakley, Tony Tebby, Steve Sutton and Chas Dillon seem impossible to emulate that you cannot possibly be a good programmer yourself. After all, the abilities of Nigel Mansell and Louise Aitken-Walker have not dissuaded anyone from taking up car driving.

SECTION ONE

This section of the *New User Guide* covers the subject-matter of the Introduction and Chapters 1 and 2 of the *QL User Guide*.

QL users need little introduction to the physical aspects of their computer systems. The essential parts of the QL system are a keyboard through which information is passed to the computer, a display screen through which information is passed back to the computer user, a box of electronics and silicon chips which process information and storage devices such as microdrive cartridges which can retain information.

In order to use your computer you must know how to connect the computer to a monitor or tv set, how to type at the keyboard, how to insert microdrives and so on. Unless you are completely new

Screen windows

Shift keys

Delete

Statements

to the QL these aspects of computing will already be familiar to you and so need not be covered here in any detail. Newcomers to the QL will find that the introductory section of the User Guide supplied with the computer is straightforward and instructive on these aspects.

This beginner's guide to *SuperBasic* begins with the assumption that you have a working QL in front of you, freshly reset and with no microdrive cartridges installed. The screen will have displayed its random dots of colour which are a side-effect of the computer's self-test and the copyright screen will have appeared.

At this point the QL awaits your instruction. Computers work at a constant speed, no matter what they are required to do, so there is no concept of a computer working hard or having a rest. Either it is working or it is not. At the moment, your QL is ticking over at 8.5 million instructions a second and its main concerns are to refresh the picture 50 times a second and to watch out for any keyboard activity. No matter what else is required of the computer, it must always take time out to perform these housekeeping duties. The instructions which require this to happen form part of the QL's operating system, which is called QDOS, the acronym for 'QL Disk Operating System'. Qdos is also responsible, among other things, for shaping and locating the characters printed on the screen.

The QL is expecting you to press the F1 or F2 key to indicate whether you want to use a high-resolution screen layout suitable for monitors, or a low-resolution screen layout more suited to tvs. Make your choice, and the screen changes to show the chosen layout of screen *windows* into which information can be placed. At first, the only area of the screen which accepts text is at the bottom of the display. This is the *command window*, where any commands you type are printed.

The QL's keyboard is immediately familiar to anyone with typing experience, although it has some additional keys which are found only on computers, and some keys which do more than do their typewriter equivalents. The *return key*, for instance, works exactly like the return key of an electric typewriter in that it forces the character next entered to be displayed at the beginning of the next line. On a computer it also has the role of indicating that a command has been typed in full and should now be actioned. This is known as 'entering' a command, so the return key of many computers, including the QL, is marked ENTER.

Keys unfamiliar to typists include the cursor keys, marked with arrows, which move the flashing cursor around the screen, and a number of additional shift keys which change the effects achieved by pressing ordinary keys. Typewriters usually have only one shift key, which changes the case of characters and gives access to the symbols on the number keys. The QL has such a SHIFT key, but it also has a 'control' key and an 'alternative' key, respectively marked CTRL and ALT. To use them, hold down any shift key (SHIFT, CTRL or ALT) and then press one of the other keys. The shift keys can be used in combination with each other, for instance CTRL-SHIFT-5, but such dexterity is not essential in order to program the computer.

The keyboard also has an 'escape' key, marked ESC, which can be pressed to stop something undesirable happening. The QL's ESC key is said to be 'soft' because, unless a software program includes specific instructions on how to proceed when the ESC key is pressed, it is ignored. Some other computers have 'hard' ESC keys which stop programmed activity whenever they are pressed.

Other 'soft' keys on the QL include the Function keys, labelled from F1 to F5 and situated on the left-hand-side of the keyboard. Without a program to give them some purpose the function keys are invariably ignored.

The QL has a 'delete' key combination which removes characters from the screen. CTRL-LEFT (hold down the CTRL key and press the left cursor key) deletes the character immediately to the left of the flashing cursor. CTRL-RIGHT deletes the character on which the cursor is flashing. The cursor's only purpose is to indicate where text can be entered or deleted. There is a way of 'turning off' the cursor so that it cannot be seen, but a few seconds of trying to cope without it is usually enough to demonstrate its value as a permanent part of any screen display.

The last key combination to be covered here is 'break', formed by holding down the CTRL key and pressing the SPACE bar. CTRL-SPACE has the effect of aborting some activity, be it the entering of a command or the execution of a program.

This guide to *SuperBasic* is devoted to the artificial and ephemeral world created within the computer rather than to the physical reality of the computer system's components. The boundaries of this world are imposed by the physical constraints imposed by the computer's design and construction, the logical constraints imposed by the QDOS operating system and the *SuperBasic* programming language, and the limits of the user's imagination. Sadly, the extent of the latter is too often exhausted long before the other, absolute, limits are encountered.

The first thing to know about the computer's internal world is that it is reactive: it does nothing until instructed to do so by the programmer. You, the programmer, decide when things should happen, in what order actions should take place, how many times activities are to be repeated and when things stop again. The computer translates your commands into instructions it can understand, and then carries them out, often using the QDOS housekeeping routines mentioned earlier.

The second thing to know about the computer's world is that everything within it is irritatingly precise. Computers do not tolerate spelling mistakes, or forgetful punctuation, or imperfect logic. By implication, precision is maintained over time. This means that if the computer produces a particular result once, it will produce exactly the same result if the exact circumstances re-occur. It is this exactness which makes computer programming useful and appealing: computing is one of the few pursuits which has a sense of absolute perfection.

The third thing to know is derived from the previous two: if the computer fouls up, it is always your fault.

At the heart of instructing the computer is the idea of the 'statement'. Just as sentences are the basic units of English expression, so statements are single *SuperBasic* entities. *SuperBasic* is

much simpler than English, and so the variety associated with English sentences is not present. Most SuperBasic statements are commands and all SuperBasic statements follow rigid rules. This actually makes programming simpler because in English we can debate whether to say 'This is different from that' or 'This differs from that', or even 'This and that are different' while in SuperBasic we can only say 'This <> That'.

Turn on your QL and try a few experiments to deduce what we can about SuperBasic. Let us try to make the computer say 'Hello'. Type in:

SAY HELLO

and press Enter. The message 'bad name' appears. The QL is communicating with us, but only to express its lack of understanding. Error message like this are not to be worried about. They are there for our benefit, to highlight a mistake on our part which the computer cannot deal with. If one appears on your screen, respond to it positively by correcting the fault command the computer has objected to.

A quick check of the manual reveals that the word SAY means nothing to the computer, but that the alternative PRINT will be understood, so try typing:

PRINT HELLO

When the Enter key is pressed the computer does not produce an error message, but the word HELLO has no meaning to it and so it has printed an asterisk. From now on it is assumed that you will automatically press Enter at the end of every command.

A further check of the manual indicates that text to be printed should be enclosed in quotes, like this:

PRINT "HELLO"

This time the computer carries out the command perfectly. Now that one set of characters has been printed it is possible to replace HELLO with any other set of characters contained within quotes and it is certain that the computer will print them as requested. The only possible exception is if the characters include additional quotation marks, because the computer cannot then determine exactly where the string of characters is supposed to end.

Words with special meaning to the computer are called *keywords*, so PRINT is a keyword and SAY is not. SuperBasic has many keywords, although it is not necessary to know about all of them in order to write programs. Every SuperBasic statement begins with a keyword.

Turning to numbers rather than text, typing:

PRINT "72"

has a predictable outcome. However, typing:

PRINT 72

has *exactly the same* effect. This is interesting, because it means that numbers are treated differently from text. Try testing the computer's mathematical skills with:

PRINT 72+16-5

The result, 83, is printed on the screen. The response to the similar command:

PRINT "72+16-5"

is quite different, though. Find out for yourself what it is.

During my Basic tutorial classes, the next programming experiment takes place away from the computer and involves pieces of paper and shoe boxes. Imagine a set of empty shoe boxes being used to store numbers of text, each box holding one number or one piece of text. To store a number, it is written on a piece of paper and stored in a box. If the number is to be used in an equation, it is examined in the box, the equation is carried out and the result can be stored in another box, or it can replace the first number in the first box. In order to indicate which box holds which number it is convenient to name them. The names of the boxes holding text have a special feature to distinguish them from the boxes holding numbers.

This primitive set-up is a close analogy of the way the QL stores numbers and text in its memory. Parts of the QL's memory are reserved to hold numerous numbers and pieces of text, with each element of that memory being given a name by the programmer. The proper term for one of these elements is 'variable', because what a variable contains can be varied by the programmer.

To place a value in a variable, a new SuperBasic command is necessary:

LET salary = 1024

Salary, of course, means nothing specific to the computer; it is just a name for a variable. Similarly, the computer does not know or care if 1024 refers to pounds, dollars or carrots. 1024 is simply a

**Print to
screen**

Numbers

**Values and
variables**

value held in the variable. When you type in this command nothing particular happens on the screen, but somewhere in the QL's memory a shoebox called 'salary' has been created to store the value '1024'.

Now the experiments with LET and PRINT can begin in earnest, but before doing so, a few more maths symbols will add some variety. Addition and subtraction are represented by + and -, as you would expect, but multiplication and division are represented by * and / in all computer languages. So here is how to award yourself a pay rise, calculate your tax and print your net income:

PRINT salary

LET salary = 1024 + 350

PRINT salary

LET tax = salary * 0.24

PRINT tax

PRINT salary - tax

It is important that the chosen variable names do not begin with a number, otherwise the QL would confuse the variable name with a value and print an error message. LET 5X = 25 might be interesting algebra but is not valid SuperBasic. LET X5 = 25 is, however, acceptable to the computer because that initial letter in the variable name removes all ambiguity.

If the screen is now getting cluttered, a useful command is:

CLS

Clear screen

CLS stands for CLear Screen, but on the QL it actually clears just one window, the one in which the results of your commands have been appearing. Notice that CLS forms a statement by itself, with no need for any number or text to follow it. PRINT can also form a statement by itself – it prints a blank line. LET, however, makes no sense to the computer or to the programmer unless it is followed by some sort of assignment. The QL objects if it finds keywords in unexpected places, such as in:

PRINT CLS

Keywords are therefore 'reserved' so that they cannot be used as variable names.

Variables, you will recall, can hold text as well as numbers provided that the variable name identifies that it belongs to a text-holding memory box. In SuperBasic, names of text variables must end with a dollar sign, such as NAME\$ or CITY\$. Examine this statement:

LET Text\$ = "Made in Britain"

The LET keyword indicates that a value is going to be placed in a variable. 'Text\$' is a valid name for a text-holding variable. The equals sign separates the variable name from its value, which is a string of characters contained in quotes. The whole command makes perfect sense to both programmer and computer. To prove that the text has been stored, type:

PRINT Text\$

Now we can return to our first attempts to type in valid commands and understand more clearly what was going wrong. SAY HELLO was clearly nonsense because SAY is not a SuperBasic keyword. PRINT HELLO was a correct SuperBasic command, but the response was an asterisk, not the word "HELLO". It is now clear that the computer understood HELLO to be a variable name, but there was no variable so-named in its memory, so it printed an asterisk instead of a value.

This month most of the rules about variable names have been covered, together with some useful keywords and some thoughts about valid and invalid SuperBasic statements. Below are a handful of statements, some of which are valid and some not. If you cannot work out which are which, your QL will help you.

Test yourself

```
LET BEER = 5X
LET NAME$ = "AMANDA"
PRINT AMANDA
PRINT 5+23 * 7/12
LET CLS = 104
PRINT NAME$
```

So far, of course, the computer has only carried out one command at a time. These are known as 'direct commands'. Next month, the SuperBasic tutorial series forms programs from lists of separate commands and introduces more keywords to the SuperBasic vocabulary.

QPAC1 and Pointers

QPAC1, released in 1988, includes the QJump Extended Environment and some applications. Ian Bruntlett gives a user retrospect.

QPAC1 is an apparently little known package from Qjump, brought out 'some time in 1988' with very little publicity. It is a set of six jobs that do trivial things for the user, plus a set of programs that make up the Qjump Extended Environment. All six jobs are well behaved and can be put on rom and/or hotkeys.

The Qjump Extended Environment (QEE) provides:

1. Proper mouse handling.
2. Preserved job windows. You can have two programs that overlap each other without one programme ruining the display of the other.
3. A sophisticated menu handling system for use by programmers.
4. Greatly improved ALTkeys.
5. The Thing system. (The Thing system is used by programmers to store something under a name in memory for later use.) I'll cover the practical aspects of Things and hotkeys later.

There is much confusion over the QEE, maybe because it has evolved over the years, and probably because it sits in the background, doing lots of work, being taken for granted. The little table below shows that the simple stage is just having PTR_GEN loaded and that the whole stage consists of having PTR_GEN, WMAN and HOT_REXT loaded.

Files loaded	Environment name
1) PTR_GEN	Pointer interface
2) 1,WMAN	Qjump Pointer Environment (QPE)
3) 1,2,HOT_REXT	Qjump Extended Environment (QEE)

When the QEE is loaded, life is much easier. When one job's windows overlap another job's windows, the job whose windows have been overlapped is considered *buried* and will be stopped if it tries to read the keyboard or write to the screen. If a buried job does not read the keyboard/write to the screen, it can carry on processing as normal. If a job wants to print messages on the screen, but still wants to carry on processing if its windows are buried, then it uses *finite timeouts*. This just means that when a job tells Qdos to do something

(eg print a message) it tells Qdos how long to wait before giving up and telling the job it could not complete the operation. An example of finite timeouts being used would be the *Qpac2* Files menu – when it is processing a list of files, it prints each filename in a little window. Because it uses finite timeouts, it can attempt to print each file name. If it fails to print the message it just performs the file operation and moves onto the next file.

Now that jobs can overlap each other, what you see on the QL screen is a pile of jobs. To move from one to another you can press CTRL-C or you can move directly to a named job (PICKing a job) via the *Qpac2* 'Pick' menu or a Hotkey System Two HOT_PICK key.

Woken Up

Later pointer programs listen to a WAKE event. When a job is woken, it is PICKed to the top of the jobs pile and told to refresh its information. A Channels menu being refreshed would have dead channels removed and new channels added to its display.

But back to the programs, described as "Trivia for your QL". This is misleading, as it provides well written desktop utilities for

pose utility. Config is easy to use, but there have been problems when attempts have been made to configure byte values – but software developers will have checked that Config works with their programs before selling them, so that should not concern the user. A nice facility for a future version of Config would be the ability to list the current configuration settings of a file.

QPAC1 contains the following jobs:

1. The Calendar. When popped up, it reads the QL clock and displays the current month, with the current day highlighted. The calendar has three sizes of window layouts: the default, medium, shows the current month, days elapsed and the clock. When the user moves to another month, the 'days elapsed' figure is updated to show the number of days between the highlighted day and the current day. The small calendar only shows the current date. The large calendar shows everything that the medium calendar plus the base date. The base date is the date taken to be the start when the days elapsed figure is calculated. You can only change the base date by changing the calendar window size to large. This is easily done.

2. The Clock. This is basically the *SuperToolkit Two* 'Clock' command bundled into a job. It shows on the date and time on screen, according to a special 'format string'. The format string may be changed by Config. Usually when you pop up a clock job you get the time from it and then go back to whatever you were doing. If you do this too often you end up with many unwanted Clock jobs in the machine. This Clock will remove itself when its windows get buried. A useful feature of Clock is date stuffing. If you are about to type the date into a letter, simply pop up the Clock, press space over its windows and it types the current date and time into your letter and removes itself.

3. The Alarm clock. This is similar to the *SuperToolkit Two* 'Alarm' command but I prefer it. The time of the alarm must be entered along with a message. You may also choose to be given a warning up to 60,

your QL, with some brass knobs on.

One of the brass knobs is a standard configuring program. The idea is that the QL software writers include a special configuration block in their programs. They would not have to write a Configure program specially for their program. The program, 'Config', looks at the special configuration block, and this tells it how to configure the software.

'Config' is usually supplied with most packages that use it. All of the QPAC1 jobs may be configured with this general pur-

45, 30, 15, 5, or 2 minutes before the alarm is triggered. There is a special message which is intended to run *SuperBasic* commands or to execute hotkeys. It works with the older version of PTR_GEN supplied with QPAC1, but it reports 'Failed to execute' when used with later versions of PTR_GEN.

4. The Calculator. This is useful. The only criticism could be its lack of extra functions/numeric bases. Despite that, it is more than adequate for normal use. To add to it may overload it with unnecessary features.

The result of a calculation may be entered into the stuffer buffer. (The stuffer buffer is a special name for ALT SPACE, as many pointer programmes put filenames and other short pieces of text there.) This is very handy when you are in the middle of some work and need to work something out. Simply pop up Calculator, work it out and place it in the stuffer buffer. Then return to where you left off and press ALT SPACE to have the answer typed in for you.

Intelligent

Calculator has a % key that is easy to use. The key strokes "200+50%=" yield "300". The precedence of operators (which mathematical operation is done first) may be changed from infix (the same as *SuperBasic* and other programming languages) to none.

5. The Typer. This is a simple program that emulates an intelligent typewriter. You type in a line of text, and it is sent to a printer. I don't know why this program was written; there are plenty of text handling programs for the QL. It seems to be an exercise in using the Pointer Environment and the Qtyp I/II spelling checking extensions. In the Quanta library there is a simple text editor, *SEDIT_BAS*, written in *SuperBasic* by Qjump. If you can remember your printer control codes, you can send them from Typer.

As an example, if you are tired of the jagged edges produced by many desktop publishing packages when printing big text on a dot matrix printer, try this (on a Citizen 120D):

1. Run Typer.
2. Type in (C)1Bh(C)1BW1LINE ONE.
3. Type in LINE TWO.
4. Type in (C)1Bu(C)1BW0LINE THREE.
(Treat (C) as SHIFT+ESC).

Alternatively you could always do it easier in *Quill*. Simply use F3DD4P33 ENTER ENTER F3JC ENTER and special printer driver with ESC,h,ESC,W,1 as preamble and ESC,u,ESC,W,0 as postamble. This works on a Citizen 120D but the control codes may be different on other printers. All it does is set the printer up to use tall and wide characters. Because the char-

acters were defined by the printer manufacturer, they take full advantage of the printer's facilities and so come out smooth.

6. Sysmon. (System monitor). This is invaluable. It provides at the top of the screen a real time display of which job is using which part of your QL's memory. That is handy in keeping track of memory fragmentation. What is even more handy is the way it continually monitors system tables and gives an audible warning if one of them becomes corrupted. It may be configured to remove itself when its windows are buried. It can't catch all crashes but is a useful guardian.

The Qpac1 jobs seem trivial at first, but they are an indispensable aid when using a computer and are well worth getting. You will find them useful.

Hotkeys are used to do something unrelated to the job you are using at the time

you press the hotkey.

There are three default hotkeys. The first, ALT ENTER, is the familiar 'last line recall' of *SuperToolkit Two*. The other two look after a 'stuffer buffer'. The stuffer buffer is a hotkey whose purpose is to type a string of text into the keyboard for you. It is a simple matter for a program to place a filename there for you so that you do not have to type it in manually. In the example boot program, there is a hotkey defined (ALT a) which types in the *SuperBasic* command:

```
HOT_DO " ":INPUT 0;_q$:HOT _STUFF
_q$ <ENTER>
```

This is very handy if I have a repetitive piece of text to be typed in, as it allows the user to *edit* the stuffer buffer.

The Hotkey System Two includes the ALTKEY procedure of *SuperToolkit Two*,

```
100 REMark *****
110 REMark * Intelligent boot for Qjump Extended Environment *
120 REMark *****
130 REMark * Conditions of use:
140 REMark * 1. SuperToolkit Two must be available
150 REMark * 2. PTR_GEN,WMAN,HOT_REXT must be available
160 REMark *
170 REMark * The following popular programmes are catered for:*
180 REMark * 1. QPAC1 Alarm, Calculator, Calendar, Clock,
190 REMark * Sysmon and Typer
200 REMark * 2. QPAC2 Qpac2 (1 file holds all its menus)
210 REMark * 3. Psion Abacus,Archive,Easel,Quill
220 REMark * 4. Lightning lng_text_ext
230 REMark * 5. Qliberated SuperBasic
240 REMark * 6. Turbo Charged SuperBasic
250 REMark *
260 REMark *****
270 TK2_EXT
280 WMON
290 LMULT 5,lng_text_ext,ptr_gen,wman,hot_rext,Qpac2
295 LMULT 2,Qlib_run,Turbo_tk_code
300 Hotkey_text
310 Hotkey2_definitions
320 HOT_GO
330 STOP
340 :
350 DEFine FuNction IIF$(cond%,_on$,off$)
351 IF cond%:RETurn _on$:ELSE RETurn off$
355 END DEFine IIF$
357 :
360 DEFine PROCedure Hotkey2_definitions
370 LOCal Prgn$,Prog_type%,Pn,fno
380 RESTORE 67
390 fno=1
400 Pout £2;'Function keys'
410 PRINT £2;'Press ALT Fn to "load" a prog'
420 PRINT £2;'Press ALT CTRL Fn to "pick" the same prog'
430 FOR Key=232 TO 248 STEP 4,234 TO 250 STEP 4
440 READ Prgn$
450 IF Prgn$<>" THEN
460 Prog_type%=Prgn$(1) INSTR 'l@'
470 IF Prog_type%=1:Prgn%=Prgn$(2 TO)
480 IF Prog_type%=2 THEN
490 Pn=Prgn$(2 TO 4)
500 Prgn%=Prgn$(5 TO)
510 ERT HOT_LOAD (CHR$(Key),Prgn$,Prgn$&'_'&Pn&'Kb',P,Pn)
520 ELSE
530 ERT HOT_LOAD (CHR$(Key),Prgn$)
540 END IF
550 IF Prog_type%=1 THEN
560 ERT HOT_WAKE(CHR$(Key+1),Prgn$)
570 ELSE
580 ERT HOT_PICK(CHR$(Key+1),Prgn$)
```

NOTE:
read # for £
throughout

and has a function based equivalent, HOT_KEY. As many people used ALTKEYs to store little lines of SuperBasic to make life easier, it was a logical progression to have the HOT_CMD variant. This CTRL-Cs to SuperBasic before typing in the text. There is only one snag – it is already obsolete! If you have commands that you will only type into SuperBasic, then put them either on a HOT_CMD or on a HOT_KEY or ALTKEY. Why? Because if you try to use a HOT_CMD within a MultiBasic job, the command ends up in SuperBasic and not the job that you intended.

Popping-up

The most common use of hotkeys seems to be the popping up of programs. With the Hotkey System Two, any QL program that can be EXECed may be popped up. All you need to do is to decide whether or not you would like the program to be resident in memory (HOT_RES) or loaded from a file (HOT_LOAD). There is a further variation which only loads or executes the program if it is not already running – if it is running, it is already 'woken'.

A waking job is brought to the top of the window pile and told to refresh its information. To do this you use 'HOT_RES1' or 'HOT_LOAD1', the '1' showing that only one copy of the program will be running at any one time. When a HOT_RES is performed, the file is loaded into the RESident procedure area if possible, otherwise it gets loaded into the common heap. If you wish to force the file into the common heap, then you use either HOT_CHP or HOT_CHP1.

The Thing system is used to control access to jobs stored in memory. When the Hotkey system is told to make a program file resident (HOT_RES, HOT_CHP), it loads it into memory and defines it as a Thing. The name of the Thing is the jobname stored within the file. If you have *Thing & Eprom Manager* by Jochen Merz then you will have the TH_LOAD (load a file as a Thing and name it) command. You use a HOT_THING key to run a thing when a key is pressed.

To pop up EXECutable files, we have used HOT_RES, HOT_RES1 et al. The equivalents for popping up Things are HOT_THING and HOT_WAKE. Pressing a HOT_THING key creates a new copy of a job even if it is already running. Pressing a HOT_WAKE key causes a job to be created if it doesn't already exist. If, when a HOT_WAKE key is pressed the job already exists, then it is brought to the top of the job pile and told to refresh its windows.

There is one hotkey type, HOT_PICK, that brings a job up without creating any new jobs. This is useful when there are a lot of jobs in the machine and you want to move quickly from one job to another. It can be a lot faster than pressing CTRL-C

```

590 END IF
600 END IF
610 PRINT £2; 'ALT F'; fno TO 10; Prgn$
620 fno=fno+1
630 END FOR Key
640 REMark *****
650 REMark * To change function key progs, edit below: *
660 REMark *****
670 DATA "Alarm", "Sysmon", "Calculator", "Clock", "Calendar"
672 DATA "@64Quill", "@64Easel", "@64Archive", "@64Abacus"
674 DATA "Typer"
680 PRINT 'ALT | Pick SuperBasic'\\
690 ERT HOT_PICK('|', '')
700 REMark *****
710 REMark * Hotkeys for Super/MultiBasics *
720 REMark *****
730 IF VER$='JSL1'
740 ERT HOT_LOAD('~', 'multib_exe'):ERT HOT_PICK('~', 'SB')
750 PRINT £2
760 Pout £2; 'Minerva hotkeys:'
770 PRINT £2; 'ALT sh' Load a MultiBasic'
780 PRINT £2; 'ALT ' Pick a MultiBasic'
790 END IF
800 REMark *****
810 REMark * THESE LINES ARE ONLY RUN IF QPAC2 HAS LOADED *
820 REMark *****
830 IF FTEST(Qpac2)=0 THEN
840 Pout £1; 'QPAC2 hotkeys:'
850 PRINT 'ALT - Turn current job into a button'
860 PRINT 'ALT . Pick all the buttons'
870 ERT HOT_THING('-', 'Button_sleep')
880 ERT HOT_THING('.', 'Button_Pick')
890 RESTORE 990
900 FOR n=1 TO 9
910 READ q$
920 ERT HOT_WAKE(CHR$(48+n), q$)
930 ERT HOT_PICK('!@£$%^&*(''(n)), q$)
940 PRINT 'ALT !CHR$(48+n) TO 10; q$
950 END FOR n
960 REMark *****
970 REMark * Define Qpac2 menus to pop up from ALT-1 to ALT-9 *
980 REMark *****
990 DATA 'Exec', 'Wake', 'Pick', 'Files', 'Rjob'
991 DATA 'Sysdef', 'Channels', 'Jobs', 'Things'
1000 ELSE
1010 PRINT 'QPAC2 not loaded'
1020 END IF
1030 REMark *****
1040 REMark * END OF QPAC2 SPECIFIC LINES *
1050 REMark *****
1060 END DEFine Hotkey2_definitions
1070 :
1080 REMark *****
1090 REMark * Procedure Loads into RESPR a MULTitude of files *
1100 REMark * from a list. *
1110 REMark * LMULT <number of files>, file1, file2, ..., fileN *
1115 REMark * LOCAL VARIABLES: *
1116 REMark * Pn set if the current prog is a Psion prog *
1117 REMark *
1120 REMark *****
1130 DEFine PROCEDURE LMULT(n, a$, b$, c$)
1140 LOCAL tot: tot=0: n=n+1: prt%=0
1150 IF prt% THEN PRINT £2; 'LMULT'
1160 FOR Pnum=2 TO n
1170 x$=PARNAM$(Pnum)
1180 IF FTEST(x$)=0 THEN tot=tot+FLEN(\\x$)
1190 END FOR Pnum
1200 addr=RESPR(tot)
1210 FOR Pnum=2 TO n
1220 x$=PARNAM$(Pnum)
1230 IF FTEST(x$)=0 THEN
1240 IF prt% THEN PRINT £2; Pnum-1; HEX$(addr, 24); \\x$
1250 LBYTES x$, addr
1260 CALL addr
1270 addr=addr+FLEN(\\x$)
1280 ELSE
1290 PRINT £0; Pnum-1; 'Programme "; x$; " missing so skipped.'
1300 END IF
1310 END FOR Pnum
1320 IF prt% THEN PRINT £2; 'LMULT usage="; tot\\
1330 END DEFine LMULT

```


many times to move past the other jobs. When a HOT_PICK is defined, a Job name is associated it rather than a file name.

The HOT_DO and HOT_STUFF commands were used in my HOT_KEY to edit the stuffer buffer:

```
HOT_DO " ":INPUT £0;_q$:HOT_STUFF
_q$
```

The HOT_DO procedure executes a hotkey from a programme without the user needing to press the key. The HOT_DO "" attempts to stuff the contents of the stuffer buffer into SuperBasic's keyboard queue. This is then read by the INPUT command, giving us a cheap and easy way of editing what is already in the stuffer buffer. Then the HOT_STUFF procedure places the edited text back in the stuffer buffer.

The Hotkey actions are performed by a standard job called 'HOTKEY'. If you delete the job by accident or design, you need not worry – just enter the command HOT_GO and you will get your hotkeys running again. If you want to delete the 'HOTKEY' job in a program or when you don't have access to the SuperToolkit Two command 'RJOB' then enter the command HOT_STOP.

If you are using an archaic piece of software that likes to think it can use the ALTkey then help is available. You use HOT_OFF to switch off a hotkey and HOT_SET to switch it back on again. If you want to switch the key off and remove the hotkey definition then use HOT_REMV. Be careful with HOT_REMV because if you have a hotkey set up for a resident program (eg HOT_RES, HOT_CHP), then any job created from that hotkey will be removed when you remove that hotkey. The HOT_SET function can also be used to move the definition of a hotkey from one key to another. For example if you wanted key '.' to behave as if it were not defined and key '/' to behave as if it had the old definition of key 'A' then you would type in ERT HOT_SET ('.', '/').

There are functions for examining the state of the Hotkey system. Of these the user will use HOT_LIST the most as it lists the Hotkey definitions. It will send its output anywhere so HOT_LIST \user will give you a hard copy listing of what the hotkeys are. The other two functions for examining hotkeys are HOT_TYPE (what does the key do?) and HOT_NAME\$ (what is the name associated with the key?).

I have been referring to various procedures and functions, all starting with 'HOT_'. How do we use them? Well the procedures (eg HOT_GO) are easy. Just type in HOT_GO. The functions return a value. Why? So that if we wanted to we could write a piece of SuperBasic to set up hotkeys and behave intelligently when given the return value. The program could check for success or failure with the assignment of a hotkey and perhaps prompt the user to insert the appropriate program disc. If we

```
1340 :
1350 DEFine PROCedure Pout(ch,tst$)
1360 UNDER £ch;1
1370 PRINT £ch;tst$
1380 UNDER £ch;0
1390 END DEFine Pout
1400 :
1410 DEFine PROCedure Hotkey_text
1420 LOCAL FN_3$
1430 FN_3$=CHR$(240)
1440 Pout £1;'SuperBasic hotkeys'
1450 ERT HOT_KEY (" ",FN_3$&"GT")
1460 ERT HOT_KEY (";",FN_3$&"GB")
1470 ERT HOT_CMD ("<",FN_3$&"S",")")
1480 ERT HOT_KEY (" ",FN_3$&"OS")
1490 ERT HOT_KEY ('a','HOT_DO " ":INPUT£0;_q$:HOT_STUFF _q$',')
1500 ERT HOT_KEY ('e','ED £2;')
1510 ERT HOT_KEY ('e','CLEAR:','ED',')
1520 ERT HOT_KEY ('k','CLS £2:LIST',')
1530 ERT HOT_KEY('o','INPUT£0;" ";$1"TO"1_q$:RENAME _$, _q$',')
1540 ERT HOT_KEY ("I","WINDOW£2;512,202,0,0:BORDER£2;1,255,"")
1550 ERT HOT_KEY ("I","WINDOW£2;256,202,0,0:BORDER£2;1,255,"")
1560 END DEFine Hotkey_text
```

just want SuperBasic to halt with an error message if a hotkey assignment fails then we use the ERT procedure. This takes one parameter, a number that is taken to be a Qdos error code. So the command

```
ERT HOT_LOAD ('0', 'Config')
will halt with the error 'in use' if the key '0' has already been defined; otherwise it will set up a hotkey to load the Configuring program.
```

The definitive guide to the 'HOT_' commands must be the Qpac2 manual. Table 1 is an index, which splits the commands up into related groups. There is an index on

Qpac2 page 30, but it does not include all the commands.

To define a hotkey, a line of SuperBasic like:

```
ERT HOT_<type> (<key>, <name>)
```

will be used. <type> is the type of hotkey being set up eg 'LOAD' and the name is simply the name of the programme to be handled; ERT HOT_LOAD ('0', 'Hexagame_exe'). When a HOT_KEY/ALTKEY is being defined, the <name> part is the actual text; ERT HOT_CMD ('E',

Page	Function	Action
EE32	HOT_KEY	Type in text
EE33	HOT_CMD	PICK SuperBasic and type in text
EE35	HOT_LOAD	Load and execute a job
EE41	HOT_LOAD1	HOT_LOAD else WAKE it
EE38	HOT_THING	Execute a Thing
EE40	HOT_WAKE	Execute the Thing else WAKE it
EE34	HOT_RES	Execute a job from memory
EE41	HOT_RES1	Execute the job else WAKE it
EE34	HOT_CHP	Execute a job from memory
EE41	HOT_CHP1	HOT_CHP else wake it

Enquire about hotkeys

Page	Name	Type	Action
EE42	HOT_NAME\$	Function	Return text linked with a key
EE42	HOT_TYPE	Function	Return a key's action type
EE42	HOT_LIST	Procedure	List hotkey definitions

Hotkey system commands

EE39	HOT_PICK	Function	CTRL-C a named job to the top of the window pile
EE44	HOT_DO	Procedure	Execute a hotkey from a programme
EE44	HOT_STUFF	Procedure	Place text in the stuffer buffer
EE41	HOT_STOP	Procedure	Start the 'HOTKEY' job
EE41	HOT_GO	Procedure	Stop the 'HOTKEY' job

Hotkey maintenance

EE43	HOT_OFF	Function	Switch off a hotkey
EE43	HOT_SET	Function	Switch a hotkey back on OR move a key definition to another key
EE44	HOT_REMV	Function	Remove a hotkey completely

'CLEAR:', 'ED', ' ').

The simple operations of the hotkey system two have been covered. When the Psion programs have to be catered for, some extra information has to be included. A flag 'p' is used followed by the amount of memory to be allocated. All hotkey commands have the ability to give the job they handle a job name that is different to the

```
ERT HOT_LOAD ('0',flp1_Quill,Quill_64Kb,P,64)
      |           |           |           |
Triggering Key   |           |           |
Program filename |           |           |
                  |           |           |
Helpful Job name |           |           |
Flag for a Psion prog |           |           |
Give it 64Kb
```

filename, you simply include the new job name after the file name:

There are other flags - 'u' unlocks a jobs window so that it can overwrite the windows of another job. This is only useful for clock programs. And there is a 'g' flag which is used to set up a window for jobs who place a window on the screen and close it before they really need to. The 'i' flag is used to deal with impure programs that modify their own code (eg *text87*, *Super/Turbocharged* progs). Impure programs cannot share their code between multiple copies of themselves. When multiple copies of a program are run, impure programs consume far more memory than normal jobs as they cannot share their code. Some programs write to the screen directly or bypass the Qdos keyboard routines by calling KEYROW directly - to make life easier, an 'f' option is provided. A job marked via the 'f' option will be halted as soon as its windows are buried and started again when its windows are uncovered.

The flags and altered job names above are not restricted to hotkeys. The EXEP command has been provided to handle jobs like hotkeys except from a SuperBasic command, eg:

```
EXEP Quill,Quill_64Kb,P,64
```

So far a lot of the setting up has relied on the use of SuperBasic. The user who knows little or no SuperBasic, and wants to get as much out of the system as possible should consider getting help by going to local Quanta sub-group meetings.

To finish I present a small, intelligent SuperBasic boot program. It is unashamedly biased towards ease of use. Here are the steps necessary to get it in action:

1. Type it in and save it on a blank disk as 'Boot', then copy these files onto it: PTR_GEN, WMAN and HOT_REXT.

2. If you have QPAC1, copy these files to the disk: Alarm, Calculator, Calendar, Clock, Sysmon, Typewriter.

3. If you have QPAC2, copy this file to the disk: Qpac2.

4. Copy these PSION files to the disk: Abacus, Archive, Easel, Quill, Abacus_hob, Archive_hob, Easel_hob, Quill_hob,

printer_dat, gprint_prt, install_dat.

5. (Ignore this step if the *Lightning SE* rom is in use.) If you have *Lightning*, copy this file to the disk: lng_text_ext.

6. If you use *Qliberated* programs, copy this file: Qlib_run.

7. If you use *Editor* then copy this file to the disk: 'extras' and call it 'Turbo_tk_code'.

8. If you use *Turbo* then copy this file to

the disk: Turbo_tk_code.

9. If you have a *Minerva* with *MultiBasic*, then copy: multib-exe

The only crucial programs are SuperToolkit Two, PTR_GEN, WMAN and HOT_REXT. The others can be added at a later date. Simply copy the appropriate file to the disc and it will be loaded on boot up.

Using the example

Having copied your files to the disc, boot your QL. The keys for loading jobs are explained by the QL itself. Here is an example session:

User presses ALT-SHIFT-F1 to load Quill.
User start writing a letter.

User wants to know what day it will be on a certain day:

Presses ALT-F5 to load Calendar.

Moves back to Quill with ALT-SHIFT-CTRL-F1

Continues with letter, returns to Calendar with ALT-CTRL-F5, etc.

There are four Quill keys defined:

1. ALT ' Goto top
- a. ALT ; Goto bottom
3. ALT , Search for text
4. ALT < Re-save document

There are seven keys for useful SuperBasic commands:

1. ALT a Edit stuffer buffer
2. ALT e Type in command for SuperBasic editor
3. ALT e Clear SuperBasic variables, etc.
Enter SuperBasic editor
4. ALT k Clear listing window & list SuperBasic program
5. ALT o Rename file
Press LAT 0 and type in old filename then ENTER
6. ALT 1 Expand listing window to full width of screen
7. ALT i Shrink listing window to its normal size

Jobs on the function keys

Press ALT-Fn to load a programme, ALT-CTRL-Fn PICKs it

- F1. Alarm
- F2. Sysmon
- F3. Calculator
- F4. Clock
- F5. Calendar

- F6. Quill
- F7. Easel
- F8. Archive
- F9. Abacus
- F10. Typewriter

(F6 to F10 is shift F1 to shift F5 on a standard keyboard)

If you want to add yet more toolkits then update the LMULT line. It is a simple procedure. Its first parameter, a number, is a count of the file to be loaded. The parameters that follow it are the filenames.

The hotkey definitions for function key jobs and the Qpac2 menus may be easily edited as I have included the job names in two data statements - one for function keys and one for Qpac2 menus. I have marked the data statements with REMark **** lines so they are easy to find.

If you want to change the names of the function key jobs just modify the appropriate name in the data statement. There are two little extras built into it. Putting @nnn (where nnn is a three digit number) at the start of a job name defines it as a Psion program and the nnn is the default amount of memory to give it. For example, consider altering the part of the statement with '@064Quill' to '@048Quill' and saving the program. The next time that you booted, the key would be redefined and pressing ALT SHIFT F1 would load Quill, give it 48Kb to play with and call it Quill_48Kb. Obviously if you want to use a key to load a different job you just edit the name in the data statement to the name of the program you want to load from the disc.

Putting ! at the start of a job name defines it as a Job that must be HOT_WAKEd instead of HOT_PICKed. This was done especially for 'Qd2', a pointer based editor by Jochen Merz.

I have marked the data statements using REMark **** etc and so they are easy to find.

If you do not want to type the program in, I can supply it on 3.5in or 5.25in disk and microdrive cartridge. Disk users should send £2 and their address. Microdrive users should send £1 and a formatted microdrive cartridge and their address. Microdrive copies are cheaper because I don't supply the microdrive cartridge. The address to send your requests to is:

QL Forum (Dept Boot Minor), 25 The Broadway, High Barnes, Sunderland, SR4 8LP.

INFORMATION

QPAC1 may be bought from Care Electronics for £21.85 + £2.30 p&p, or Jochen Merz (in Germany) for £17.50 + £2.10 p&p.

Pointer tools

Early versions of PTR_GEN and WMAN are on the Quanta SPECIALS_0 disc. Also in the Quanta library on SPECIALS_3 are the pointer environment tools by Oliver Fink.

Cwm Gwen Hall Services have various discs of public domain pointer tools from Germany.

Laser Printing from Text⁸⁷

The first thing to realise is that you are not obliged to have a special printer driver in order to use a laser printer with *text⁸⁷*.

If your laser printer has an emulation mode that is compatible with your existing dot-matrix printer, you can carry on using the driver for the dmp, setting the laser to the emulation mode. Unfortunately, not all lasers have dmp emulation modes so, if you are in the process of buying a laser, add the appropriate emulation mode to your list of requirements. The obvious dmp to emulate is the Epson FX80, that being the accepted standard in the dmp market. Most common dmp printers are 'FX80-compatible' to a fair degree, which means they will work with drivers written for the FX80. Provided your laser is also FX80-compatible, it should be possible to print from it without making changes to your existing set-up.

Figures one and two are of text printed by an Epson GQ-5000 laser printer using the existing driver for the Kaga-Taxan KP-810 dmp. That driver was in fact written for the Canon PW1080A, which is essentially the same machine as the KP-810, and is Epson-compatible. Compared to the 9-pin dmp output, there is an immediate 'quantum leap' in print quality. Laser printing is simply in another league. If you use your

Bryan Davies has been putting *text⁸⁷* through his new laser printer and finds that it is easier than you might expect.

QL for business purposes, the improvement in print quality will certainly give a more favourable image to customers.

Emulation

There are limitations on how far a laser can emulate a dmp, the most obvious one being the difference in available fonts and pitches. A typical dmp will print in character pitches of 5, 6, 8.5, 10, 12 and 17.1 character per inch; a basic laser printer may provide only 10 and 12 cpi, with possibly some other pitches in the range 5-6 and 15-20 cpi. One point to watch is that the printer will start printing in the font set as its default, unless the requested

character-pitch necessitates another font. For example, if the printer defaults to 12-pitch Prestige and the first typestyle requested by the *text⁸⁷* document is 10-pitch Pica, the printer may use 12-cpi Prestige. On the other hand, if the printer defaults to 10-cpi Courier, and the Elite SW typestyle is requested, the printer may squash the Courier to give 12-cpi (which doesn't look anywhere-near as good as 12-cpi Prestige). It is possible the driver will be made to get around this difficulty, however.

Range of texts

Figure one shows the range available from the Epson GQ-5000, using the

This text has been printed from *text⁸⁷* using the standard PW1080A dot-matrix printer driver. The printer being used is the Epson GQ-5000 laser, set to its FX80-emulation mode, with 10-cpi Prestige as the default fount. The current Typestyle selection is Pica SW, for which the printer uses the Prestige 10-cpi fount. Changing the Typestyle **now** to Elite SW, the printer switches to Prestige 12-cpi. When you select Typestyles with character pitches which do not exactly match those available in the printer, the latter makes an "intelligent guess" at the nearest of its own founts to what you have requested. For example:

Condensed Typestyle is reproduced by the printer as Modern PS (which is quite close to the 17.14 of a DMP).

Pica DW becomes Prestige 5-cpi.

PS SW (proportionally-spaced Pica) prints as PS Prestige.

Pica It SW becomes italic Prestige 10-cpi.

This is Elite It SW Bold Typestyle, reproduced as italic bold Prestige 12-cpi.

You can't utilise EDP 13-cpi or Prestige 20-cpi because the DMP driver does not include any Typestyle of those sizes.

Figure one: the type range available from the Epson GQ-5000

This text has been printed from *text*⁸⁷ using the standard PW1080A dot-matrix printer driver. The printer being used is the Epson GQ-5000 laser, set to its FX80-emulation mode, with 10-cpi Courier as the default fount. The current Typestyle selection is Pica SW, for which the printer uses the Courier 10-cpi fount. Changing the Typestyle now to Elite SW, the printer switches to Courier 12-cpi. When you select Typestyles with character pitches which do not exactly match those available in the printer, the latter makes an "intelligent guess" at the nearest of its own founts to what you have requested. For example:

Condensed Typestyle is reproduced by the printer as EDP 16.66-cpi (which is quite close to the 17.14 of a DMP).

Pica DW becomes Courier 5-cpi.

PS SW (proportionally-spaced Pica) prints as PS Courier.

Pica It SW becomes italic Courier 10-cpi.

This is Elite It SW Bold Typestyle, reproduced as italic bold Courier 12-cpi.

You can't utilise EDP 13-cpi or Prestige 20-cpi because the DMP driver does not include any Typestyle of those sizes.

Figure two: the type range printed in 10-point Courier

PW1080A driver, when the printer default is 10-cpi Prestige; **figure two** is of basically the same text printed with the default changed to 10-cpi Courier. From reading the manuals with the GQ, I hadn't expected to be able to get as many different pitches as proved possible, in both Courier and Prestige

GQ-5000 driver

A plus-factor is the ability to print in landscape mode as well as portrait. Most dmp printers can print only with the paper 'upright', whereas the laser can rotate the whole text 90°. This enables you to print, for instance, spreadsheets in normal-sized characters, instead of having to resort to condensed print. Additionally, with a bit of effort, you can construct newsletters in a format slightly smaller than A5 by printing two pages on each side of a landscape A4 sheet. With *text*⁸⁷, you can do this with only

one layout, using two columns and printing them side-by-side on landscape A4. This is shown in **figure three**. Read the text on this illustration for information on settings.

*Software*⁸⁷ now offers two printer drivers specifically for laser printers, one for the Hewlett-Packard LaserJet Series II and III, and the other for the Epson GQ-5000. The HP LaserJet printers have become the *de facto* standard in the laser world. Most other makes have a PLJ emulation mode, but be careful to check that it is the appropriate mode – some older printers offer LaserJet, LaserJet+ and other earlier variants, but the current common one is the LaserJet II. There must be some differences between these emulations, although I have found that graphics printed in LaserJet+ mode come out alright with the printer set to LaserJet II emulation. Unless you have the GQ-5000 (or compatible Epson models), you need the HP LaserJet

driver. The Epson models can use that, but they will not deliver all their built-in features with it.

Point sizes

The driver I tested was the one for the GQ-5000. Presumably, this will work satisfactorily with the GQ-3500 and the EPL-7100 also, although there will be some (hopefully minor) differences in output. The driver is loaded into *text*⁸⁷ in the usual fashion; check to make sure enough memory has been allocated to the program, because the driver is about 64 KB in size. If the driver appears to load in about two seconds, there isn't enough memory available and your previous driver is still installed. Make attachments in the usual way, to get the screen characters to match the ones on paper, if so desired. A major advantage of the GQ over the HPLJ is the

In order to obtain the impression of two pages abreast on one A4 sheet of paper, you can use two Columns in the Layout setting. The current Layout limits the Text width in each column to 90 mm. And the Left Margin is 0 mm. The Ruler Left Margin setting of 10 mm. ensures both that the first column text is moved sufficiently away from the paper edge to look acceptable and that there is a reasonable "gutter" between the two columns. The selected Typestyle is Condensed; with the Epson GQ-5000 default fount set to Courier 15-pitch, the fount actually used for printing is EDP, and the pitch is 15 cpi. The printer is set for Landscape printing.

Although *text*⁸⁷ nominally does not allow printing to a width greater than the 203 mm. maximum that Layout displays, you can "fiddle" the Ruler Margin settings to allow printing over a somewhat greater width in Landscape mode than would be possible in Portrait mode. This would allow the creation of a narrow newsletter, for instance. The printing of mini instruction books – so beloved of some software suppliers – would be possible in this manner.

For this second Column, the Layout and Ruler settings are unchanged. The overall width, from the left edge of the paper to the right edge of the text in this column, is 21 mm., about 30 mm. more than can be obtained with the paper in the Portrait position. There is one little oddity. The first line of the first & third paragraphs wanted to print to the left of the rest of the text, and Spaces had to be used at these points to bring the first line back into alignment with the rest. This spoils the justification.

Reminder -- this example was printed with the Canon DP printer driver, on a GQ-5000 laser printer. No special driver was required. However, without the special laser printer driver you cannot use the scalable founts which are built-into this printer. This also means that there is no sans-serif fount available above a size of 7.2 points (about 0.1 inch), which is the size of the original print of the example. The scalable serif and sans-serif founts which can be used with the laser printer driver permit characters to be printed up to 72 points (1 inch) high.

Figure three: instructions for printing two columns on landscape A4.

SAMPLE TIMES-TYPE TYPESTYLES:

BOLD 6-POINT.

Italic 10-point.

Normal 14-point.

Italic Bold 24-point.

Bold 32-point.

72-pt.

SAMPLE HELVETICA-TYPE TYPESTYLES:

Italic 6-point.

BOLD 11-POINT.

Normal 18-point.

Italic Bold 32-pt.

Bold 48-pt.

72-pt.

Figure four: some large sizes in Times and Helvetica

presence of two scaleable fonts, variants of the Times and Helvetica standard fonts. The printer itself will allow the size of these two fonts to be set by the user in fractional point sizes between 2 and 240 points (1 point is about $\frac{1}{72}$ in in height). The supplied driver limits this to 6-72 points, but that will be sufficient for most purposes; not many people need printed text higher than 1 in or lower than $\frac{1}{12}$ in.

Point sizes

The driver provides a fixed set of point sizes, and the user is not free to set half-point sizes, but the selection is adequate. It is 6, 8, 10, 11, 12, 14, 18, 24, 28, 32, 48 and 72 points, for both the Times- and Helvetica-style fonts. **Figure four** shows some of the sizes for both of these fonts. The Times style is typical of that used in newspapers, and has a 'classy' look about it, but the serifs (the bits at the ends of characters) can make reading a bit difficult as the size gets smaller; Helvetica is preferable from a readability point-of-view at small point-sizes, because it is sans-serif – there are no fiddly bits at the ends of characters. While the tendency in amateur dtp-style work seems to be to use the most complicated font available (such as the Old English ones frequently supplied), you don't (often) see this kind of thing in professional output. Times and Helvetica cover most requirements and are, therefore, the two most common additional fonts supplied with laser printers.

You have to be prepared for some lengthy waits when using scaled fonts, because the printer does not store anything other than the outlines of these fonts and has to create the required sizes at print time. That is, the full description of each character of, say, 72-point Times is not kept in the printer

firmware; all that is in there is a general description of the Times font, together with the provision to create any size and enhancement (within the specified ranges) of that font on demand.

The enhancements are only Bold and Italic/Oblique, but they have to be created at print time, just as sizes do.

How many scaled sizes can be created depends upon the available memory in the printer, and you might find it either refusing to create further sizes or replacing already-created sizes with ones that are currently required; the printer can be set to take either of these courses, when faced with shortage of memory. The message is, don't go overboard with the number of scaled fonts and sizes you use in one document.

The fixed-size ('bit-mapped') resident fonts in the GQ-5000 are shown in **figure five**. They are Courier 10 cpi (12-point), EDP 13 cpi (7.2-points), EDP 16.7 cpi (7.2-point), Modern Proportionally-spaced (10-point) and Prestige 20 cpi (16-point). Apart from Modern and Prestige, each of these fonts is supplied in landscape as well as portrait form; you need landscape forms of any fonts which you want to use when printing in landscape mode.

Courier has both a Bold and a Draft form, but you can use the bolding command of text⁸⁷ to get bold with all fonts. In theory,

this generated bold may not look as good as that from the built-in Courier bold fonts but, in practice, there was nothing obvious in the printed output to indicate whether the driver uses the built-in or a generated bold for Courier. There were some selection problems with the early version of driver supplied to me, but they will almost certainly have been ironed out by the time you read this. See if you can spot the errors in **figure five**.

Developments

At this point in time, you cannot access some of the GQ facilities, such as text rotation. If the demand (that is, the income) is there, no doubt there will be further developments along the laser theme from Software⁸⁷. Perhaps the most impressive thing about using the GQ driver was the total lack of fuss in getting it to work; it wasn't necessary to do anything, other than load the driver into text⁸⁷, create documents, and print. My printer is set to default to GQ mode, because that gives the best results, so all that was necessary there was to switch on. Now, more than ever before, the onlooker won't be able to tell whether the printed output is generated by an expensive office computer, or the cheap QL.

GQ RESIDENT BIT-MAPPED FONTS:

Courier 10-cpi/12-pt., Italics, Bold, Bold Italics.

EDP 16.7-cpi/7.2-pt., Italics, Bold, Bold Italics.

Prestige 12-cpi/10-pt., Italics, Bold, Bold Italics.

Prestige 20-cpi/16-pt., Italics, Bold, Bold Italics.

EDP 13-cpi/7.2-pt., Italics, Bold, Bold Italics.

Modern PS 10-pt., Italics, Bold, Bold Italics.

DBQL

Many new QL users are relative newcomers to computer programming and are interested in data programming. DBQL is a simple relational database written in Superbasic, using procedures throughout, and will run on an unmodified QL. In the first of several articles, author Tom Ashcroft introduces the core program.

Filing systems are nearly as old as written language itself and as long ago as 2000 BC, Sumerian merchants were recording their transactions on clay tablets baked into tiles to form what must have been a truly Fred Flintstone card index system. Modern card index systems probably have not advanced a great deal in principle since those days and can be regarded as the simplest form of database, which is simply a name for a store of information.

Card index systems can deal with very varied types of information but their drawback is that it is often difficult to extract information from them. They will usually be stored in alphabetical order of one of the key items: surname, say, in a name and address file, but if you wanted to find all the entries of people living in Newcastle, you would have to check every card individually and make up a list as you went along. This is laborious by hand but it's exactly the sort of thing a computer is good at doing and explains why database handling is a major area of computing.

Card index

A computer file is usually built up in a very similar way to a card index file. It is made up of records, each one of which would correspond to a single file card. Within each record, the information is arranged in a sequence of sections — name, address, phone number, say — which are called fields, each of which has a heading or field name. New records are added in the same structure, rather like filling in a pre-printed form.

Where the computer database scores, however, is in sophisticated facilities for extracting information from the files. Even the simplest commercial programs can quickly carry out searches which would take hours to do by hand and many packages, such as *Archive* or the *dBASE* series, have their own programming lan-

guage which can be used to interrogate the files and produce complex reports on the information retrieved. A database which keeps all the information in one file is known as a flat-file database, but more complex programs can link different files together through a common field and are known as relational databases. These can achieve much more complex and useful analysis of data.

Flat file

The rest of this article will describe a SuperBasic program to create and use a flat-file database. Later articles will add facilities for sorting and indexing of files, working with two or more open files to make a true relational database, adding procedures to program the database for specific applications and lastly adding a text editor and handling large text fields.

The program consists of a set of SuperBasic procedures. Many commercial programs have an elaborate control system to run the database, with pull-down menus and single key-press choices, and it would not be too difficult to program similar facilities in SuperBasic, but it would make the program much bigger and more complex for very little gain in convenience. SuperBasic has the very useful property that procedures can be called directly from the keyboard and the whole program is designed to be used in this way. The procedure names work like commands and, of course, ordinary SuperBasic commands such as DIR and CLS are also available.

A simple file has to be able to store records, accept input of new records, search the records for specific information, display the information, update records and delete records no longer required. The SuperBasic construct for storing data is the string array (see Chapter 13, page 74 of the *Sinclair QL User Guide*). Briefly, a string array is a numbered list of

strings of defined maximum length, which is set up using the DIM (dimension) statement. A two-dimensional array consists of numbered groups of strings and we will use a two-dimensional array to store our data, with each group of strings corresponding to a record, and each string in the group to a field. It is not possible to have a mixed string and numeric array and numerical data must be stored as strings and converted to numbers only when required for calculation. SuperBasic coercion makes the conversion easy, in either direction. The size of an array is limited by the memory available but even an unexpanded QL has ample memory for a useful database. The array will be called A\$ for simplicity.

To create the database, we must first decide the number and size of the fields required. If, for example, we wanted a file to hold details of members of a chess club, we would need fields to hold members' initials, surname, address, town, postcode, date of joining, subscription and so forth. It is best to work out the fields with pencil and paper before starting to program but **listing one** accepts as many field names as required in an open-ended way to allow for last minute changes of mind, and then sets up appropriate array.

Field names

Field names are input in 2350 in a loop, fname. Names up to 10 characters long are allowed and longer names are truncated by 2370 while shorter names are padded with spaces by 2380. Names are stored temporarily in X\$ by 2390, which also increments the counter, fields, to keep a tally of the number of names entered. When all the names have been entered, inputting (Shift/ESC), exits the loop and 2410 asks for the length of the longest field. Line 2420 ensures it is at least 10 characters. Unfortunately, all the elements in an array are dimensioned to the same

length, however much is used by the fields allotted to them, so the whole array has to be dimensioned to the size of the longest field likely to be encountered. In the case of an address file 25 characters per element would be ample for most entries. Line 2430 creates the array, with space for 50 records initially. SuperBasic arrays are always dimensioned with a hidden record (A\$(0)), which is a perfect place to store the field names, and the loop beginning at 2440 slices the names from X\$, easy when they are all 10 characters, and enters them in the appropriate elements of A\$(10). Finally, 2470 sets the record counter, nn, to 0, line 2480 sends a message to the screen and the database is ready to receive information.

Listing two accepts data from the keyboard and enters it in the array.

Copyright

The core of the ENTER procedure is a FOR...ENDFOR loop which accepts the INPUT of an entry for each field in turn (1210-1250). This loop is nested within a REPEAT loop, Entryloop, (1190 to 1270), which increments the record counter, nn, and brings up new records until it is exited by entering a single copyright sign (Shift/ESC) as a field entry. On each pass through the loop 1260 checks to see whether the array is completely filled and if so saves the file to microdrive and reopens it with space for 10 more records. When the entryloop is exited, nn will have been incremented in preparation for the record which was aborted in 1230, and 1280 decrements it again to the value of the last record. The procedure opens by clearing the screen and typing instructions in channel 0 and ends by clearing everything from the screen.

Slow search

So far, our information is held only in memory where it is vulnerable to total loss if the computer should crash, and it ought to be written to permanent storage as soon as possible. In SuperBasic this is done by opening a channel to a microdrive file and printing the data to the channel, usually as part of the QUIT procedure of the database (**listing three**).

Line 1920 checks the resave flag and aborts the save if this is not set. The flag is only set if a record is changed or a new record entered. The SSAVE procedure is now called. Line 1990 checks the oldfile flag and if it is set then makes a temporary copy of the existing microdrive file with a different name and deletes the old file. This is to ensure that if a computer crash should occur during the saving process at least the previous version of the file is still available on microdrive. When the file is next loaded the program will need to know certain parameters to DIMension the ar-

ray, namely the number of records (nn), the number of fields (fields) and the maximum number of characters per field (chars). So when the new microdrive file is opened by 2000 under the original filename, these variables are first printed to the file. Then two nested FOR...ENDFOR loops read all the records in A\$ into the file, record by record and field by field. Line 2030 checks each record to see if it is marked for deletion and if so skips to the next record without filing the marked one.

The procedure ends by closing the file, deleting the temporary file and returning to QUIT which prints up a message on the screen, with the ENDING procedure. After a short pause, all variables are deleted from memory by the CLEAR command (**listing four**).

The process of opening a file is really the reverse of saving it. The procedure OPEN (**listing five**) simply requests a name for the file and calls the LLOAD procedure to get the information from microdrive. Line 2190 INPUTS values for nn, fields and chars and 2200 DIMensions an array to a size ten records but avoids blocking large amounts of memory in an unexpanded QL. The procedure then reads from the microdrive into the array field by field until nn records have been input. Finally the file is closed and the oldfile flag is set.

We now need facilities to retrieve and display information from the file, and this is the area in which customised display procedures can be written to suit particular files. This will be discussed in more detail in a later article and for the present the display procedure simply prints the fields of the record in sequence on the screen.

The record to be displayed is the current record and its number is held in the variable, c. Once again a FOR...ENDFOR loop is used to print each field to the channel number stored in the variable, chan. The same procedure is used to send information to the printer (**listing seven**). The startup procedure will have opened channel 3 to SER1. When chan = 1, output is to the screen; when chan = 3, it goes to the printer. Field names are printed by 3050 unless suppressed by the flag nemo.

Searching for records is a topic to be explored further in a later article. The simplest way to search for a record is just to page through the file and the three short procedures in **listing eight** enable this to be done.

These have all been given one-letter names for quick entry from the keyboard. G for GOTO (a reserved word in SuperBasic) requires a number to be passed to it and sets the record pointer (c) equal to that number before calling DISPLAY. To see record 30, for instance, you would type

```
g 30 <ENTER>
```

and it would appear on the screen. Typing g nn <ENTER> will display the last record in the file. N and B move the record

pointer to the Next record forward or back respectively in the file. If the top or bottom of the file is reached they print warning messages on the screen.

An automated search is provided by the group of nested procedures in **listing nine**.

First, a search item is requested and c is set to 0. The second procedure, CO, CONTINUE SEARCH, sets the search baseline pointer, d, equal to one more than the current record number and the nested FOR...ENDFOR loops (1470 to 1510) use the INSTRing function to compare the search item successively with each field of each record until a match is achieved or the end of the file is reached. The INSTR function is used because it will find the target string even when it forms only part of the contents of the field and because it is independent of upper or lower case. For example, "mit" will match with "Smith" or "Mitchell".

Environment

This method can be rather slow when searching a large file and if the search key is only likely to be found in one field, such as a surname, restricting the comparison to this field gives a very fast search. The following changes will make this modification.

Delete lines 1480 and 1500.

Add the following lines

```
1342 FOR j=1 TO fields:PRINT a$(0, j)
1344 INPUT "Enter the number of the field required ", fld
```

When a match is found in INSTR_SR, 1490 sets the current record pointer, c, to the record number, the found flag is set, and the program returns to procedure CO which calls the DISPLAY and stops. Calling CO directly from the keyboard continues the search and displays the next matching record. If the end of the file is reached without a match or further matches, c will have been incremented equal to nn+1 and 1410 prints the appropriate message depending on the value of the flag, found.

Once a record has been located it can be amended or if necessary deleted altogether from the file (**listing ten**).

To amend a record, a FOR...ENDFOR loop prints each field in turn on the screen and waits for a new string to be INPUT. Keying <ENTER> leaves the field unchanged while any new string typed will replace the original one. Lastly, the changed record is displayed for approval (**listing eleven**).

A record selected for deletion is not actually removed from memory. It is marked with a flag which makes the QUIT procedure leave it out when the file is next saved

to microdrive. Line 1550 places a "D" in the hidden field 0, sets the resave flag and increments the variable, deleted, which holds the number of deleted records currently in the file. A message on the screen tells that this has been done.

Bare Bones

The foregoing procedures all the facilities needed to set up and use a simple database. All that is now required is a 'menu' of the commands available and a screen environment to work in. **Listing twelve**,

START and MENU, provide this. They also set initial values for some variables.

The START procedure begins by opening a channel to the printer and then opens a screen channel (#4) for the menu, adjusting the size of #1 to suit. The window dimensions are designed for mode 8, tv mode, and the menu layout would have to be adjusted to look well in mode 4. After setting initial values for the flag variables the MENU procedure is called to print a list of commands in #4. Procedures are called by typing the first two letters of the command followed by <ENTER>. Next, Back and Goto need only the first letter typed, for

ease of paging.

These procedures provide the bare bones of a database management system, but lack the safety features that a commercial package would provide. For example there are no checks on validity of inputs and accidentally entering a letter when the program is expecting a number will produce a prompt error message. But one of the strengths of the open procedure plan of the program is that an error of that kind does not matter. Just call the procedure again and do it right the second time!

The next article will deal with sorting and indexing files.

For @ read © throughout.

Listing one

```
2280 REMark *****CREATE
2290 DEFine PROCedure cr
2300 CLS
2310 INPUT "Please enter file name: ";f$;resave=1
2320 x$="":fields=0
2330 CLS#0: PRINT#0,"Enter field names in order,
up to 10 characters in length.\"Enter Shift/ESC
to quit."
2340 REPEAT fname
2350 INPUT "Field name ";fields+1;" : "; field$
2360 IF field$="@ THEN EXIT fname
2370 IF LEN(field$)>10 THEN field$=field$(1 TO 10)
2380 IF LEN(field$)<10 THEN field$=field$&FILL$
(" ",10-LEN(field$))
2390 x$=x$&field$:fields=fields+1
2400 END REPEAT fname
2410 INPUT "How many characters in the longest
field?"chars
2420 IF chars<10 THEN chars=10
2430 DIM a$(50,fields,chars)
2440 FOR j=1 TO fields
2450   a$(0,j)=x$((j-1)*10+1 TO (j-1)*10+9)
2460 END FOR j
2470 nn=0
2480 CLS:CLS#0:PRINT "The file ";f$;" is open for
entries"
2490 END DEFine cr
```

Listing two

```
1150 REMark ***** ENTER
1160 DEFine PROCedure en
1170 CLS:CLS#0:resave=1
1180 PRINT#0," Type information field by
field\" Key ENTER at the end of each
field\" Shift/ESC to quit"
1190 REPEAT entryloop
1200   nn=nn+1:CLS:PRINT "Record no ";nn
1210   FOR fld 1 TO fields
1220     INPUT(a$(0,fld)&" : ");q$
1230     IF q$="@ THEN EXIT entryloop
1240     a$(nn,fld)=q$
1250   END FOR fld
1260 IF nn=DIMN(a$) THEN PRINT "Saving to
microdrive. Please wait.":ssave:lload
1270 END REPEAT entryloop
1280 nn=nn-1
1290 CLS:CLS#0
1300 END DEFine en
```

Listing three

```
1910 REMark ***** QUIT
1920 DEFine PROCedure qu
1930 IF NOT resave THEN ending:RETurn
1940 ssave
1950 ending
1960 END DEFine qu
1970 REMark ***** SAVE
1980 DEFine PROCedure ssave
1990 IF oldfile THEN COPY "mdv1 "&f$ TO "md
v1 "&f$&"_temp":DELETE "mdv1 "&f$
2000 OPEN NEW #6,"mdv1 "&f$
2010 PRINT#6,nn-deleted:PRINT#6,fields:PRINT#6
,chars
2020 FOR record=0 TO nn
2030 IF a$(record,0)="D" THEN NEXT record
2040 FOR fld=1 TO fields
2050 PRINT#6,a$(record,fld)
2060 END FOR fld:END FOR record
2070 CLOSE#6
2080 DELETE "mdv1 "&f$&"_temp"
2090 END DEFine ssave
```

Listing four

```
2500 REMark ***** ENDING
2510 DEFine PROCedure ending
2520 CLS:AT 10,9:PRINT "The file has been saved":PA
USE 200:CLEAR:CLS
2530 END DEFine ending
```

Listing five

```
2100 REMark ***** OPEN
2110 DEFine PROCedure op
2120 CLS:INPUT "Enter file name: ";f$
2130 lload
2140 CLS:AT 10,7:PRINT "The file "&f$&" is open
":PRINT#0 TO 13;nn;"
2150 END DEFine op
2160 REMark ***** LLOAD
2170 DEFine PROCedure lload
2180 OPEN_IN #6,"mdv1 "&f$
```

Late corrections

Listing two: 1240 a\$(nn,fld)=q\$
 Listing five: 2140 ... TO 13;nn;"records"
 Listing six: 3030 ... THEN PRINT "Deleted": ...
 Listing nine: 1520 END DEFine instr_er
 Listing twelve: 1060 ... 0:nemo=1:chan ...


```

2190 INPUT#6,nn:INPUT#6,fields:INPUT#6,chars
2200 DIM a$(nn+10,fields,chars)
2210 FOR record=0 TO nn
2220 FOR fld=1 TO fields
2230 INPUT#6,a$(record,fld)
2240 END FOR fld
2250 END FOR record
2260 CLOSE #6:oldfile=1
2270 END DEFine lload

```

Listing six

```

3000 REMark ***** DISPLAY
3010 DEFine PROCedure display
3020 IF chan=1 THEN CLS
3030 PRINT"record ";c;:IF a$(c,0)="D" THEN
PRINT (Deleted)":PRINT:ELSE:PRINT:PRINT:END IF
3040 FOR fld=1 TO fields
3050 IF NOT nemo THEN PRINT#chan,a$(0,fld);"; ";
3060 PRINT#chan,a$(c,fld)
3070 END FOR fld
3080 END DEFine display

```

Listing seven

```

1580 REMark ***** PRINTER
1590 DEFine PROCedure pr
1600 chan=3:display:chan=1
1610 END DEFine pr

```

Listing eight

```

1620 REMark ***** GOTO
1630 DEFine PROCedure g(x)
1640 c=x:IF c>nn THEN c=nn
1650 display
1660 END DEFine g
1670 REMark ***** NEXT
1680 DEFine PROCedure n
1690 c=c+1
1700 IF c>nn THEN PRINT\ TO 10;:PAPER 2:PRI
NT" end of file ";:PAUSE 50:PAPER 0:c=nn
1710 display
1720 END DEFine n
1730 REMark ***** BACK
1740 DEFine PROCedure b
1750 c=c-1
1760 IF c<1 THEN PRINT\ TO 9;:PAPER 2:INK 7:PR
INT" start of file ";:PAUSE 50:PAPER 0:c=1
1770 display
1780 END DEFine b

```

Listing nine

```

1310 REMark ***** SEARCH
1320 DEFine PROCedure se
1330 c=0:found=0:CLS
1340 INPUT"Enter item for search ";s$
1350 co
1360 END DEFine se
1370 REMark ***** CONTINUE SEARCH
1380 DEFine PROCedure co
1390 d=c+1:IF d>nn THEN PRINT\ TO 9;:PAPER 2:PRI
NT" end of file ";:PAPER 0:RETurn

```

```

1400 instr_sr
1410 IF c=nn+1 THEN :IF found THEN :PRINT "NO
FURTHER RECORDS FOUND":ELSE :PRINT"NOT FOUN
D":END IF :RETurn :END IF
1420 display
1430 END DEFine co
1440 REMark ***** INSTR_SR
1450 DEFine PROCedure instr_sr
1460 c=nn+1
1470 FOR record=d TO nn
1480 FOR fld= 1 TO fields
1490 IF s$ INSTR a$(record,fld) THEN c=reco
rd:found=1:EXIT record
1500 END FOR fld
1510 END FOR record
1520 END DEFine instr_st

```

Listing ten

```

1790 REMark ***** AMEND
1800 DEFine PROCedure am
1810 CLS:CLS#0
1820 AT#0,0,1:PAPER#0, 1:PRINT#0," ENTER";:PA
PER#0, 0:PRINT#0," leaves item unchanged "
1830 PRINT#0," To change an item, type new ver
sion"
1840 resave=1
1850 FOR fld=1 TO fields
1860 PRINT a$(c,fld)
1870 INPUT q$:IF q$="" THEN :NEXT fld:EL
SE :a$(c,fld)=q$:END IF
1880 END FOR fld
1890 display:CLS#0
1900 END DEFine am

```

Listing eleven

```

1530 REMark ***** DELETE
1540 DEFine PROCedure de
1550 a$(c,0)="D":resave=1:deleted=deleted+1
1560 PRINT"Record ";c;" is marked for deletion"
1570 END DEFine de

```

Listing twelve

```

1010 REMark ***** START
1020 DEFine PROCedure start
1030 OPEN #3,ser1
1040 OPEN #4,con_448x30a32x16
1050 WINDOW 448,170,32,46
1060 resave=0:oldfile=0:nemo+1:chan=1:dele
ted=0:c=1
1070 PAPER 0:PAPER #0,0:PAPER#4,0:INK 7:I
NK #0,7:INK#4,4:CLS:CLS#0
1080 menu
1090 END DEFine start
1100 REMark ***** MENU
1110 DEFine PROCedure menu
1120 CLS#4
1130 PRINT#4,"CReate OPeN ENter Next Back SEarch
Goto AMend PRinter QUIT Continue Delete CLS"
1140 END DEFine menu

```


SOFTWARE FILE

INFORMATION

Program: QL Quick Posters

Supplier: Dilwyn Jones Computing, 41 Brom Emrys, Tal-y-Bont, Bangor, Gwynedd.

Price: £10.00. Available on microdrive, 3.5in or 5.25in disk. Please supply a formatted cartridge with microdrive orders. Suitable for an unexpanded QL.

It came as quite a shock to me to find that I had been only using half the facilities my Star LC10 printer was capable of. Dilwyn Jones's new program, *QL Quick Posters*, certainly reaches the parts of the printer which other programs don't.

QL Quick Posters was designed to make text-only posters quickly, simply and effortlessly by harnessing facilities built into many modern printers and it certainly lives up to its promise. It is *not* suitable, however, for the following printers:

Centronics GLP, Brother M1009/M1109, Epson FX80/RX80/MX80, 9 pin Amstrads, MT80/81, OKIM80 series, and Panasonic KXP1080.

It IS suitable for most 24-pin dot matrix printers (eg Star LC24-10), some laser printers with Epson style control codes and some of the more recent 8 pin dot matrix printers (eg Star LC10).

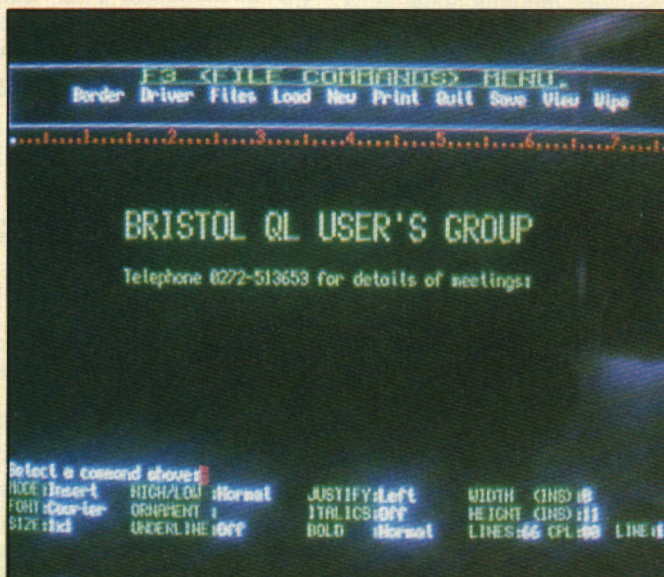
If you are uncertain as to whether your printer is suitable, then give Dilwyn a ring first.

In essence, the program uses printer facilities such as the following:

Single, double, triple and quadruple text.
Up to eight fonts at a time.
NLQ or draft printing.
Up to four ornaments (eg

QL QUICK POSTERS

This program stretches your printer, reports John Shaw.



shadow, outline or both).
Proportional spacing.
Justification (eg left, right, centre, left and right).
Italics.
Subscripts.
Superscripts.
Bold and double strike.
Underlining.

Don't worry if your printer doesn't have all of these facilities, you can still use the program omitting the absent ones. My Star LC10, for example, has no ornaments but has all the rest.

Dilwyn has, very thoughtfully, provided printer drivers for the following printers: Star LC10, Star LC24-10 and Epson GQ3500. There is also a printer

driver configured program so that you can adapt the codes to suit your particular printer. Very clear and comprehensive instructions are included in the 12-page manual which accompanies the disk.

What of the program itself? Well, it can be loaded very quickly by use of the normal BOOT or indeed it can be multitasked under something like *Taskmaster*.

Once loaded, it shows a display similar to *Quill*. A top section shows the command instructions, the large middle section is for the text input and there is a small window at the bottom showing you which page size, font size, text size etc. you have chosen.

Use of the program could not be simpler. Press <F3> (Commands) and you will get the following display:

Border Driver Files Load New Print Quit Save View Wipe

BORDER allows you to put a page border of text characters around the outside of the page. Different ones may be specified for the corners and sides.

DRIVER: the program loads the default driver when BOOTing up, however, this facility allows you to change drivers mid-stream.

FILES is a DIRectory listing command.

LOAD loads a page previously SAVED by the program. It will not load other files ie doc or lis.

NEW defines a new page. The default size is 80 wide by 66 deep (about fanfold sheet size). A small adjustment is needed for A4. Sizes may be varied to suit paper and print size.

PRINT: choosing this option brings up a sub-directory with the following options:

NQL/DRAFT?

PRINT PITCH? 1 2 3 4 From 12 to 20 characters per inch PROPORTIONAL SPACING?

PRINTER JUSTIFICATION? NUMBER OF COPIES (1 - 99)?

QUIT takes you out of the program

SAVE saves a page in the Quick Poster format. An A4 page takes about 6K of disk space or 12 mdv sectors.

VIEW is a very useful device which gives you a preview of the poster layout. It allows you to compose your masterpiece without leaving too many ugly gaps.

WIPE clears the page without altering the size settings.

Press <F4> (Styles) and you will see:

Bold Font High/Low Italics Justify Ornament Size Underline

BOLD gives you the option of Bold or Double Strike

FONT gives you the choice of all the fonts available on your printer.

HIGH/LOW gives sub-

scripts or superscripts according to choice.

ITALICS uses your printers italic facility.

JUSTIFY: the options are left, right, centre or left and right.

ORNAMENT: these are special effects such as shadowing or outlining. Up to four can be used.

SIZE: up to four sizes of text are allowed in any one document. My Star LC10 does 1x1, 1x2, 2x2 and 4x4.

UNDERLINE is the standard printer underline function.

<SHIFT F4> puts you into Overtyping mode.

It is worth mentioning that when I used the following two

combinations on my Star LC10: NLQ/Italic/Bold/Double Strike in double and quadruple sizes, the printout was corrupted. All other printer combinations were perfect. This fault turned out to be in the printer, not in the software.

Dilwyn also includes some sample pages with the program: CAR STICKER_page shows how to compose car stickers (for your Group?). SMALL_EXAMPLE_page prints a page the size of a postage stamp. DISC_LABEL_page provides the ideal size for your floppy disk printing. EXAMPLE_page is an unashamed advert for Dilwyn's products in many different fonts and sizes.

So here we have a well-made product capable of about 32,000 print style combinations and the whole thing designed so that even the most inexperienced QL user should be printing posters in minutes. The manual is as always with Dilwyn's products, comprehensive and easy to follow, and the whole product represents excellent value for money.

EDIT MODE	REFRESH	QL QUICK POSTERS	TYPESTYLES	COMMANDS:
		© Dilwyn Jones, June 1990	MENU :	ESCAPE:

BRISTOL QL COMPUTER GROUP?

RING CHRIS GREGORY ON 0272-513653

Sinclair QL ... The computer that won't lie down and die!

MODE:Insert HIGH/LOW :Normal JUSTIFY:Centre WIDTH (INS):8
FONT:Courier ORNAMENT : ITALICS:Off HEIGHT (INS):6
SIZE:4x4 UNDERLINE:OFF BOLD :Both LINES:36 CPL:80 LINE:24

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 - 4100 Duisburg 11 - W-Germany
Tel./Fax 0203/501274

QL-Emulator for ATARI ST

Emulator (Hardware, QJUMP's Drivers (including WIN), AtariDOS and more...) £163
EPROM Cartridge for ST (makes machine autoBOOT) £29.50
Floppy Adaptor to connect QL-Disk-drives to the ST £8.90
Other ATARI adaptors, serial & parallel cables, Harddisks etc, at low prices!

QL-Hardware

ABC-Keyboard Interface (for IBM PC/XT keyboards) £41
QIMI - Internal Mouse Interface with Real-Time-Clock £39.90
High Quality Mouse for QIMI or ATARI ST £26

Software

QD - NOW VERSION III! Many additional features! Write for details £36.50
Upgrade from Version I - £10 Upgrade from Version II - £20

OPTR - Re-released & Updated! Toolkit for Machine Code and SuperBASIC Pointer Environment Programmers. Revised manual and keys/macros updated. CONFIG is also explained £29.50

Update (new manual & disc) £13.50

QMENU Many QD-Users will know it: the Menu Extension. Now with more facilities, even real subdirectories are treated in a new file-select window. Programming instructions and examples for SuperBASIC and machine code. £10.50

DIAMONDS This is definitely a 'just one more go' game to keep you up at night. £10.50

QSHANG

QL version of the well-known game. Nice graphics, 1 or 2 player game and more. £15

Patience Strategy Card game, runs under Pointer Environment. £13.50

Super Games Pack: Pengi, Arcanoid, Firebirds, Doppel Ion & Ion Gold. £25

EASYPTR Create your own Pointer Environment Menus and sprites and use them in your own SuperBASIC or machine-code programs. Many users waited for this package, as creating Menus is not very easy - until now. Put all the items you need on screen - that's it! Supports all the PE facilities, even split sub-windows and also uses the Menu Extension! Many examples. £39.00

QLQ £20 Thing & EPROM Manager £18 Firebirds £10.50
Brain Smasher! £12.00 QSUP £26 Arcanoid II £10.50
Ion Gold & Doppel Ion £10.50

Please add £2.10 for postage and package. All programs except Arcanoid, Firebirds & Ions need memory expansion. 3.5" discs only! All prices excl. V.A.T. E&OE

The Advanced Wordprocessor for the Sinclair QL

text⁸⁷ version 3.00 and text⁸⁷*budget*, offer today's state-of-the-art user-friendly environment for document production. Extremely fast in operation with documents of any size and capable of mixing and displaying different type styles and sizes both on the screen and on paper. Support multiple columns, multiple rulers, and multi-line headers and footers. With integrated spelling-checker in three languages and editable dictionaries. Budget version cannot be used with 24-pin, laser and deskjet printer drivers. Both versions come with 34 sophisticated drivers for dot-matrix and daisywheel printers. You can add the extra drivers for state-of-the-art output not available from any other QL program.

fountext⁸⁸ and founded⁸⁹, graphic printer-driver and editor for text⁸⁷ provide high quality graphic founts without the limitations in text editing and document size imposed by QL desktop-publishing programs. More than 40 quality founts of up to 72 pixels in height are supplied and new founts can be designed or captured from saved screen images. For Epson and compatible 9-pin and 24-pin printers.

24⁸⁸, text-mode printer drivers for selected Epson LQ, NEC P, Star LC and XB and Panasonic KXP 24-pin printers offer multiple typefaces, proportional spacing, micro-justification, double height, shadow and outline modes (depending on the printer model).

text⁸⁷: £60 ● text⁸⁷*budget*: £45 ● fountext⁸⁸ + founded⁸⁹: £40
24⁸⁸: £15 ● typeset⁹⁰ :deskjet: £20 ● laserjet: £40 ● GQ5000: £40

Send for our new comprehensive leaflet

Software available in English, French, German and Italian. Prices are inclusive of airmail. Payable by cheque, Postal Order or Eurocheque. Please specify language and disk system (all 31/2" and 51/4" formats can be supplied). text⁸⁷ requires at least 256K memory expansion.

Software⁸⁷, 33 Savernake Road, London NW3 2JU

Technical Tips

HARD DISKS

It would perhaps be better if 'hard' disk drives were to be referred to as 'fixed' rather than 'hard', as that conveys the design better. In fact, some large computer systems have 'fixed disks' of a basically similar style to those we use with micros. Whatever the name, the magnetic disk upon which data is stored cannot be removed from the drive by the user. (There is, however, a category of hard disk which is removable as a complete unit, and reference to this is made later in this article.) Although the basic hard disk drive unit is likely to be the same whatever micro it is used in, there are appreciable differences in the controller units and the driver software. Some of the comments that follow do not at present apply to hard disks used with the QL.

To have large storage capability in a small space, with a conventional rotating-disk drive, mechanical tolerances need to be fine. You can't get the requisite tolerances with a disk that can be 'got at' by the user, so the disk is hidden inside a sealed casing.

Several

It isn't just one disk; as might be expected, once you've constructed the basic mechanism for driving one disk, it doesn't take a lot more space to provide several disks. Likewise with the read/write head mechanism: as with a double-sided floppy disk drive, each disk has a head on either side. Three disks will have six heads, and that is about the minimum configuration you will meet. You can have several times that many disks and heads.

In physical size, there is not much to choose between floppy and hard drives. Each can come in one-third, one-half or full-height form. Hopefully, you won't see too much of full-height drives; they are about 10 cm high. Some of these, typically

Bryan Davies, proud owner of a hard disk, charts some of the advantages and pitfalls of relying on this ultra-fast method of data storage.

with 10 MB capacity, will be found for sale quite cheaply at computer fairs. Half-height drives are about 5 cm high, and this is the standard height for the 5 $\frac{1}{4}$ in drives you will see in most micro systems. Third-height usually applies to 3 $\frac{1}{2}$ in drives; being correspondingly smaller than half-height, the drives are often mounted in frames to fit half-height slots, so no space is saved. Internally-fitted third-height hard drives which do not use standard 5 $\frac{1}{4}$ in frames are a similar size to bare 3 $\frac{1}{2}$ in floppy drives.

There are other reasons for sealing the disk inside a casing. The fine tolerances make it essential to keep dust out, as far as possible. If you take a look inside a PC casing after even a few weeks use in a typical office environment, you will be amazed at the amount of dust that has accumulated. There is a vicious circle here; hard disk drives get hot, so a fan is needed to cool them, and the fan sucks in dust. While the QL collects dust on top without getting very dirty inside, anything with a fan in it gets distinctly dusty inside.

Head Crash

You may have heard the expression "head crashes", although it was more associated with large fixed drives in the past. The read/write head has to be very close to the disk (magnetism weakening rapidly with distance), and there is a significant risk of the head actually touching the disk surface, especially if dirt particles come

between head and disk. In floppy drives, the heads do touch the disk, and so does dirt, which is why disks eventually give errors, however well treated.

In hard drives, use is made of a "boundary layer effect", where a thin film of air is more-or-less permanently in contact with the disk, and any reasonably light object suspended above the disk floats in this 'cushion' of air.

A similar manifestation of this effect occurs with an aircraft when it is about to touch down on a runway. At a certain level above the runway, the aircraft can get into a floating state where it drifts on along the runway without coming down, unless some positive action is taken to force it down further.

The disk drive head is permanently in this state, floating a minimal distance above the disk. As there is nothing positively locating the head, a sizeable knock can cause it to crash down onto the surface. This is why you are advised not to move, or bang, a computer containing a hard disk drive, and why there are 'parking' mechanisms to lock the heads when they are not in use.

The basic operating principles of hard and floppy drives are similar, but the hard disk spins at 3600 rpm, as against 300 rpm for a floppy disk. The more surface that moves past the heads in a given time the greater the data transfer rate. In an ideal situation, the rate at which data can be passed to, or taken from, the disk would match the speed of rotation, with data stored in a simple sequence on the disk.

In practice, the disk sectors

tend to go past the heads at a greater rate than data can be transferred. This leads to the use of "interleaving", as described in *DIY Toolkit* last month. Sectors are not read/written consecutively, but skipped after each read/write operation, to allow time for one lot of data to be digested. The drive electronics wait until one revolution is completed before taking the next sector, but that would waste time. An interleave factor of 3 (1:3) is usual with floppy disks, with two sectors skipped after each one accessed.

Faster

While the hard disk can transfer data faster than the floppy can, the much higher rotation speed causes the same situation, and an interleave of 3 has been the norm until recently. With the advent of different methods of transferring data, faster processor chips, and the use of disk "caches", the interleave has been reduced to 2 or 1 in the past few years (mainly with higher-capacity, expensive, drives).

Returning to disk rotation speed, there is a basic difference between floppy and hard drive here, because the floppy drive does not run unless requested to by software, whereas the hard drive runs continuously. There is a significant start-up time with floppies, but none with hard disk, and data access speed is proportionate to this.

Hard drive sectors contain more data than floppy ones do – typically two KB against 0.5 KB. This gives a higher data transfer rate, but reduces the number of files that can be stored; any file requires at least 2 KB of space. Disk caches are used to reduce the amount of input/output operations with hard disk by storing some of the data transferred in a separate memory area, in the expectation that part of that infor-

mation will be required again immediately.

Programs are written such that the same sections of code are frequently accessed over and over again; if that code can be obtained from ram rather than disk, much time is saved. It is fairly normal to find that disk read/write operations are reduced by around 70% when a sizeable cache is used. Put another way, the benchmarked speed of disk access can be increased by a factor of 2-3. The cache can be in the hard drive controller electronics, or can be in a section of the main computer ram memory, or in both places.

Caching

If caching is partnered by faster encoding forms, the performance gap between floppy and hard disk becomes much greater. The encoding method used with floppies is mfm (modified frequency modulation, or double-density; fm, or single-density, was an earlier method), and that is still used for some hard drives. It was superseded by rll (run-length limited), which effectively gave a 50% increase in capacity to the drive.

SCSI (small computer systems interface) gives a much-greater data transfer rate, but it tends to be used on the less-common micros, such as the Macintosh, whereas esdi (enhanced small devices interface) is used on PCs, and gives a similar transfer rate, in the range 5-15 mbps (million bits per second). The nominal, physical access time to data on a hard disk is usually quoted in ms (milliseconds), with up to 20 ms being quite good and 70 ms now being considered slow.

The effective access time for a conventional hard disk on a PC can now be reduced to microseconds, by using a DPT (brand name, not encoding method) controller. For comparison, the access time with a floppy drive is more like 100 ms, with a data transfer rate of around 250 Kbps (thousand bits per second).

Some of the foregoing comments may have given the impression that hard disk activity

is based on using one disk at a time, but there would not be much sense doing this in a device which has multiple disks and heads. The disks are stacked one above the other.

Cylinder

You can imagine the tracks on each disk to be extended up and down to the parallel disks, such that each track – when extended to all the disks – is called a 'cylinder'. There are a number of concentric cylinders (equal to the number of tracks), and it is quite possible to read/write data to the same track number on any of the disks at the same time, because all the heads move together, and are therefore located above the same track numbers. A smart drive controller can spread one file over several disk surfaces in this way, but there obviously has to be a routine contained within the controller to specify how data is to be read after having been written in this fashion.

Just where the 'controller' is varies. Traditionally, there has been a separate pcb on which all drive control circuitry was fitted, but the recent trend has

been to incorporate much of the circuitry with the drive itself, and put the remainder with the main processor, on the 'motherboard'. Much as this trend may please the makers, and give improvements in data transfer rate, it does mean that it is gradually becoming impossible to 'mix-and-match' controllers and drives in different computers, and this restricts the user's freedom of choice and increases the cost.

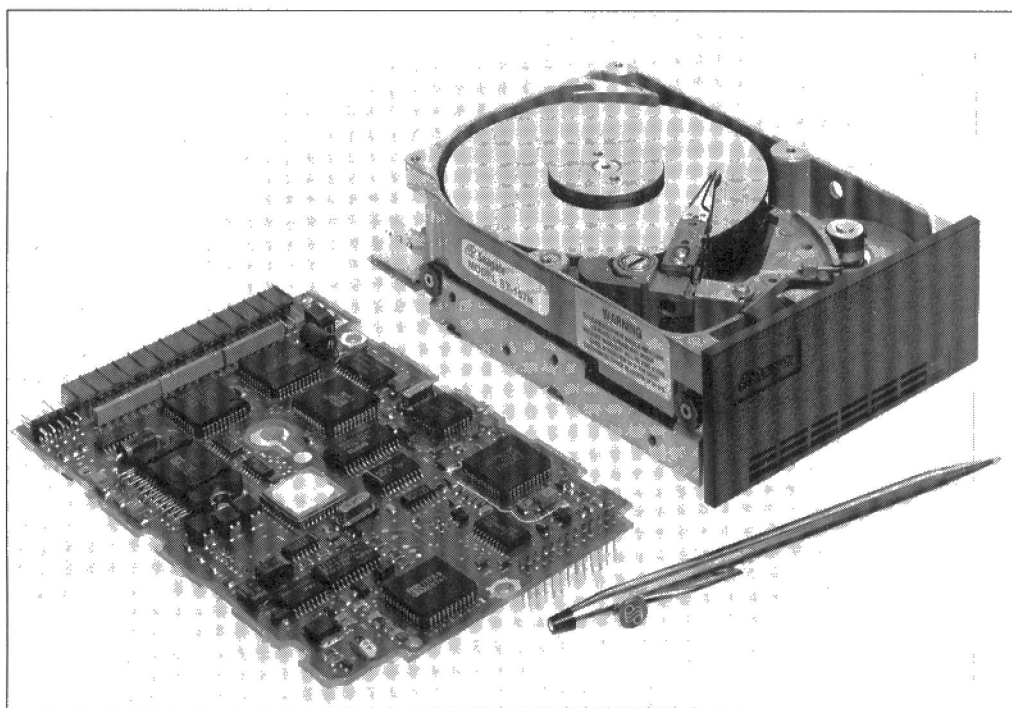
There are some drawbacks to hard disk drives. They hold a lot of data, but that means a lot can be lost. Users may want to move data from one computer to another, without the aid of physical interconnections, and the typical hard drive can't be moved easily. The heat problem was touched on above, and it brings another problem with it – noise. A computer with a hard disk is usually noisy, possibly irritatingly so, because of the drive mechanism and cooling fan. Hard drives take a fair amount of current, and a hefty power supply is required for them. A major factor is the need for software specifically to cope with both the input/output of the hard drive and the way file storage is structured on it.

Having got a device with seemingly-massive storage capacity, how do you go about covering yourself against the possible loss of some or all of the data? Conventionally, you back everything up to floppy disk! Back where you started. If you've got a hard drive, you've almost certainly got at least one floppy drive also, so backing-up this way is a possibility, but at what cost?

Backup

30 MB of files on hard disk may require 50-100 floppy disks for backup; you can't use all the space on each disk, and large files in particular will cause much space to be "lost". That number of floppies costs serious money, but the time taken to copy the files – usually several hours – may be much more serious. A good utility program can reduce the space and time taken, dramatically; for example, down to 10-15 high-density disks and 10-15 minutes for 30 MB. But the program adds some more to the cost.

An alternative method is to use a "tape streamer" – some form of tape-based recorder/player which will allow all the data from a hard drive to be put



The inside of a hard disk mechanism (not for the QL)

HARD DISKS

on (maybe) one tape. This avoids the user having to be present to pop disks in and out, and considerably reduces the storage space needed for the backups. Floppy disk and tape allow the data to be transferred, by hand, to other systems. Some companies (notably Tandon) manufacture hard drives which can be plugged in and removed quickly, and are small and rugged enough to be transported by hand, or through the mail. Very convenient, but rather pricey.

Reformat

Whatever you do, *don't* kid yourself there is no need to backup hard disk files; maybe nothing will go wrong with drive or controller, but it is quite within the bounds of possibility that you will delete files yourself by mistake, or even reformat the drive completely. Hard disk "disasters" are quite common.

Many users know well enough the frequency of errors with removable media, particularly microdrive cartridges. Generally speaking, there

should be less errors with hard disk than with floppies, but my own experience (and reading) suggests the reverse if true. For me, errors with floppies are almost non-existent, but problems of some sort with hard disk are too frequent for comfort.

This is not to say that the basic drives are not more reliable – they very likely are – but the associated *software* can lead to more situations requiring user intervention, to avoid data loss. Software which performs memory management tasks can play some odd tricks, including falsely indicating that a hard disk is giving errors. You tend to lose sight of just how much data you actually put onto hard disk, and it is easy to think of error situations in absolute terms, rather than related to the number of files on the drive, the number of transfer operations, etc. Still, each occasion when you have to stop what you are doing, in order to deal with a drive-related problem, is time wasted. Please note that, as yet, experience with hard disk in QL systems is limited, and the comments made here

are based mainly on experience with other computers, but the basic factors apply whatever the computer.

You can choose to treat hard disk simply as a very large floppy, storing all files together, in one great mess. However, once you get up into the hundreds of files, keeping track of them becomes a major problem. QL software is still relatively easy to track, in so far as most main program files have no, or few, associated files. With *The Editor*, you have the *Edt_hlp* and *Edt_charset* files, both of which you can choose to omit, but you can't do without the *Xtras* file (or its equivalent). The files names are usually recognisable.

Fount files

But what about fount files, for instance? If you have *SpeedScreen*, *Lightning*, *Professional Publisher*, *text⁸⁷* etc., can you tell which fount files are associated with which program? The obvious QL utility programs tend to display file names in batches of maybe 8-

12, and it can take an age to go through a whole hard disk-full of files.

For many users, it is essential on hard disk to batch files into meaningful categories. On PCs, you can use sub-directories on floppy disks, but the directories themselves take up space and, presumably, few users will have anything other than the root directory on their floppies.

QDOS on the QL has no need of sub-directories, since there is really no space for them on microdrive cartridges. This means that the basic software structure for handling sub-directories is not there but, on hard disk, some degree of sub-directory structure is a necessity. If you don't feel the need, maybe you don't really need to use hard disk. One final point – don't think of buying hard disk for your QL until you have some extra memory, because the hard disk sub-system is likely to require a fair bit of the basic 128KB memory itself and you may end up with insufficient space to run any significant programs.

★ sinclair QL ★

EEC LTD MAIN SUPPLIER OF Sinclair QL COMPUTERS & PRINTERS QLs FROM £125 PRINTERS FROM £130

The expandable system for small businesses, beginners, and experts.

QLs COMPLETE, JM FULLY TESTED AND WITH 6 MONTHS WARRANTY
TV lead, QL Software 2.35, Quill – Word Processor, Abacus – Spread Sheet, Archive
Records, Easel – Business Graphics, JM Rom £150

£125

Customers buying one of the QL Computers are given one year's free membership to
QUANTA

(Help, Newsletters, & 400+ Library Programs – most free).

QUANTA MEMBERS can obtain a £5 discount

"The JM QL can run all Programs available for the QL system."

★ SPECIAL OFFERS ★

while stocks last

NEW PHILLIPS 9CM073 PROFESSIONAL COLOUR MONITOR

14" E.G.A. Enhanced graphics, 85 chars,
RGB Input, 0.31 D/P, with tilt and swivel
stand and QL lead ready to plug in and go
(RRP £379.00)



ONLY
£200
inc VAT

UNIVERSAL DISK DRIVE

1mb, 3.5in cased, complete with built-in power
supply, mains switch & 13amp plug, external
dip switches adapt drive for QL, PC, Atari,
Amiga, etc, comes with full instruction book,
plastic cover, and free DS/DD disk. P&P £6.00

£75 inc. VAT QL lead £10

TWO DRIVES £130 lead £15

UNCASED DRIVES

NEC FD1036A 1mb 3.5in 1/3 height

£35 inc VAT

Epson version with 5/8" adaptor plate

£40 inc VAT

QL Lead for disk 1/face £12

PC 102 KEYBOARDS FOR QL £25 Interface for above £75 Together £95

MANNESMAN TALLY MT81 EPSON, PC and QL COMPATIBLE £130
130 CPS DRAFT, 24 CPS NLQ. Tractor feed. Send for printouts & QL details
Interchangeable plug-in interface modules for Star NL-10 printer, Serial, Parallel/Centronics,
IBM/PC, or Commodore each £20

Prices include VAT. Terms CWO, Access or Visa. Minimum order £10.
Carriage £9.00 for printers & QL (overseas £20.00) other items £3.00 (overseas £6.00)

EEC LTD

18-21 MISBOURNE HOUSE, CHILTERN HILL, CHALFONT ST PETER, BUCKS, SL9 8UF
Tel: 0753 888866 Fax: 0753 887149

TF Services

ASTRACOM Modem

Intelligent buffered modem with text status
messages. Hayes protocol, parallel printer
port (can be used as 6k serial to parallel
printer buffer). 240vac or 9-12vdc. Autodial/
answer and many programmable registers incl
printer logging of RX and/or TX data.
BT approved. Can be used with any computer.
AC1000 (021/23 v1.5a) £175
AC1000T (021/23/22 v2.03 + tone ding) £257
AC1000 (021/23/22/22bis v2.03) (tone) .. £299
V22/V22bis upgrades ring for prices

Qualsoft Terminal

Viewdata/VT52/VT100 QL TERMINAL EMULATOR
'Een deluxe communicatieprogramma van Kwalsoft'

Multitasking for electronic mail, PRESTEL etc,
CET download. Phone directories for ALL (yes ALL)
modems for the QL, autodial (where poss) and
logon, two-way FILE TRANSFER to ATARI ST, IBM PC,
Psion Organiser (via comms link), XMODEM, real
time clock/timer, buffered logs to file/printer/
transmit file, editable command line, Swedish/
Norwegian/Danish options, translates, editor etc.
SOLVES ALL PACKAGED MODEM SOFTWARE PROBLEMS
Unbuffered modems usable with Miracle Modaptor

Software (3.5" or mdv) & manual £30

COMPUTER CLEANER

Much improved mains filters. 40-80 db cut
& 3 spike filters. 390 joule cut (3 & 4 way)
Certified X/Y 240ac capacitors

2-way (5a adaptor) £14
3-way (5a adaptor) £18
4-way (13a trailing socket) £24

All prices include VAT and postage in the UK

QL REPAIRS

QLs tested with Thorn EMI test rig
and ROM software
(MDV hardware extra if required)

£25 - 6 month guarantee

QL SPARES

Faulty QL boards (excl plug in ICs) £9
Keyboard membrane £7 Circuit diags £2
68000 cpu £8 1377 PHL £3
J1 rom sel £12 Power supply £12
8382 ULA £10 8381 ULA £10
8849 £7.50 MDV ULA £12
Other components (sockets etc) pise PHONE

BULLETIN BOARD

GBBS - The first UK QL scrolling bulletin
board. Now even larger with international
ECHOmail and 10mb of files (including
the WHOLE King James' bible)

071-706 2379 V01/23/22/22bis 24hrs

FILE TRANSFER

Programs to transfer text AND programs
error free XMODEM between computers

Available for IBM PC and compatibles,
ATARI ST and Sinclair QL. Connects to
Psion organiser via Psion comms link

Qualsoft program (per m/c) £7.50
Serial lead (name 2 computers) £10
Complete package for 2 computers £25

MAIL ORDER ONLY



12 Bouverie Place, London W2 1RB (tel: 071-724 9053)

SCROLLING BBS FOR ORDERING/QL HELP - 071-706 2379 (all speeds to V22bis)

Fax: 071-706 2379 Telex: 265451 (ref: 72/HK638645) Prestel: 017249653



SOFTWARE FILE

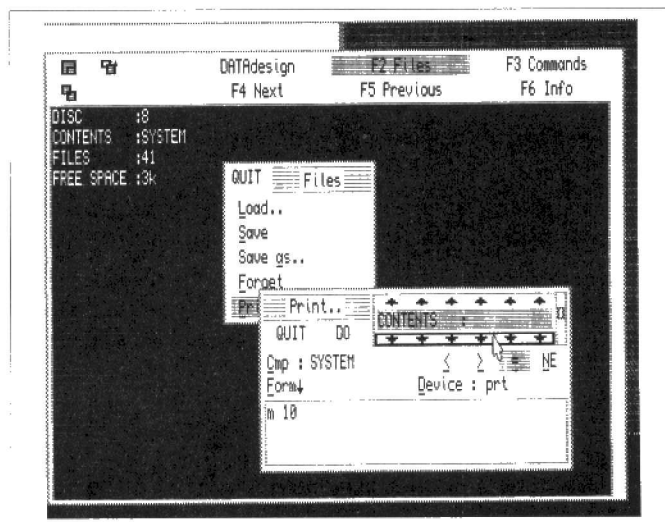
DATA DESIGN

This is another of those fairly rare items in our pages, a QL program written outside the UK. The proximity of Belgium to the English coast has made it easy for users in both countries to go to fairs and Quanta meetings in the other country, with the result that Belgian enthusiasts hear about new developments from the UK quickly. The same goes for Germany, which is just across the border and has an active QL community. The Van der Auwera brothers have been active on the QL scene for some years, but their programs have not been readily available in the UK until quite recently. *The Painter* was a joint production of the brothers, and is now well established; *Data Design* is the work of one brother, Joachim.

It is a database program, written with the stated aims of being user-friendly, allowing multiple variable-length lines in fields, having a powerful sorting routine and incorporating a printer-driver that is independent of printer model. The introductory letter with the review copy states that, apart from the field lengths being variable and speed of access being good, an important feature is that "the program only keeps that amount of memory that it really needs, and it can expand it without a sigh". This last feature should indeed produce a sigh of relief from some users!

The program makes use of the WMAN, HOT_REXT and PTR_GEN files familiar to users of the QJump products QRam and QPac, and a menu extension file from Jochen Merz. *Data Design* can be added into an existing Qram/

■ Bryan Davies tries a new database from Belgium



Qpac environment, and multi-tasked in it. There may be conflicts in other set-ups; the boot routine locked my system up when the usual other programs were present (*text⁸⁷*, *Q Switch*, *Files 2*, *Perfection* etc). The main program file is relatively small, taking less space than its companion extension files, but the total amount of memory used varied from about 130 to 200 KB, and depends upon factors such as the size of the window used and of the database loaded. Users having a Qimi or SuperQ mouse will be able to access commands with the mouse, but the keyboard can be used when no mouse is available.

The review copy was version 1.00, dated August 1990 and, with the usual progress of programs, there is a good chance that a later version is now available. There are references in the instructions to use on the Atari ST QL emulator, so I assume the program works on that.

Instructions are printed, on thirty-seven A5 pages. There is not all that much reading to do, since the pages are mostly less

than half-full. For most users, getting started will require some preliminary reading, as the program initially presents you with a pointer, and a menu panel which is simple and clear but not self-explanatory.

The left end of the menu panel contains three icons, for which I found no explanation in the instructions. They will be familiar to QJump enthusiasts, but other users might never figure them out. Two of them allow the window in which the program runs to be moved or resized; the CTRL-F4 and CTRL-F3 keyings can also be used for these operations. The latter are mentioned in the instructions, without it being pointed out that these keyings have to be followed by appropriate movement(s) of the cursor, and then pressing ENTER. Simple maybe, but not necessarily obvious to many users.

The action of moving the cursor outside the window and pressing ENTER, in the process of discovering these functions, caused the window to disappear completely and the program had to be re-

booted; it became obvious later that the CTRL-C keying will "revive" the window, but that wasn't stated (I was expecting some HotKey to be needed). The same thing happened when the remaining icon was pointed to and ENTER pressed. This icon command appears to be for bringing a multi-tasking program from the background to the foreground, a function which is not relevant when *Data Design* is used on its own. Icons are an interesting device; much effort is put into designing them so that they are self-explanatory, but they can fail completely in this job, even with an experienced user. The three used here were quite clear, after I'd discovered what they are for. Maybe a picture is not always worth a thousand words?

Selection of menu items is straightforward, and there should be a method to suit most users. If you have a suitable mouse, you can use the standard approach of moving the on-screen pointer and clicking the mouse button when the appropriate item is indicated. Alternatively, you can move the pointer with the cursor keys and press the Space bar or ENTER key to select an item. The initial menu choices are also selectable by pressing F2-F6. F6 is obtained by pressing SHIFT-F1 and gives information (name, length, number of records and fields) on the current database. F4 and F5 are the normal next/previous record command keys, once a database is loaded; an interesting feature of F5 is that it takes you back to the last record once it has reached the first once, cycling continuously if held down. F2 and F3 produce drop-down menus, and the items on these menus are selectable by

INFORMATION

Program: *Data Design* V1.00

Price: £BEF 3000; £55 by cheque; £45 by card or postal account

Supplier: Progs, Haachstraat 92, 3008 Veltem, Belgium. Tel. (0103216) - 488952

pressing the underlined letter, usually the first one of the named item. For example, L is pressed to Load a file. This approach is common in the micro world in general, but is often not implemented in QL programs; those who have experience of it will almost certainly regard it as the sensible way to go about selection from the keyboard.

Another sensible menu feature, which is common elsewhere but not on the QL, is the "greying-out" of items which cannot currently be selected, leaving only the available items clearly shown. The use of Space and ENTER interchangeably is not universal; when you get down to the menu level of choosing a file to load, for instance, ENTER is required. On sub-menus, items for which there is a further sub-menu level are indicated by ... after them (eg Forget ...); in some systems, a right-pointing arrow is used for the same purpose.

QJump

The program writer seems to have assumed that any purchaser of the program will already be familiar with the QJump way of doing things, and the instructions do not explain how to use the pointer environment. This is both a pity and rather confusing. There are facilities which a user could find very handy, but which he/she may never realise exist. Lack of explanation can also lead to frustration. An instance is the F2 Files menu item; having selected this, and followed it by L (to load a file), you are automatically presented with a directory of database files on the default device and can move the pointer to a desired file and press ENTER (not Space) to load it. Should you want to load a file from another device, the way of doing this is not clear. The pointer initially sits in a box with a double arrow pointing to the left. It is not explained that pressing Space or ENTER gets you nowhere, unless you move the pointer slightly to the left, whereupon a box appears around the name of the default device. Pressing Space now allows you to edit the device name; pressing ENTER brings up a sub-menu offering mdv1_ mdv2_ flp1_ flp2_ win1_ and win2_ as di-

rectory devices. Files with a particular extension can be listed, if the appropriate extension is typed into a small box labelled "Ext:". Although this utility area is subsidiary to the database program, it can be very useful, if the user knows how to work with it. The smoothness of operation of the window/pointer environment, and the way windows can be removed to reveal underlying windows looking "untouched", is impressive, though it involves a lot of key/mouse actions. There is plenty of "white space" in the instruction book; it is a pity some of it wasn't used to explain the

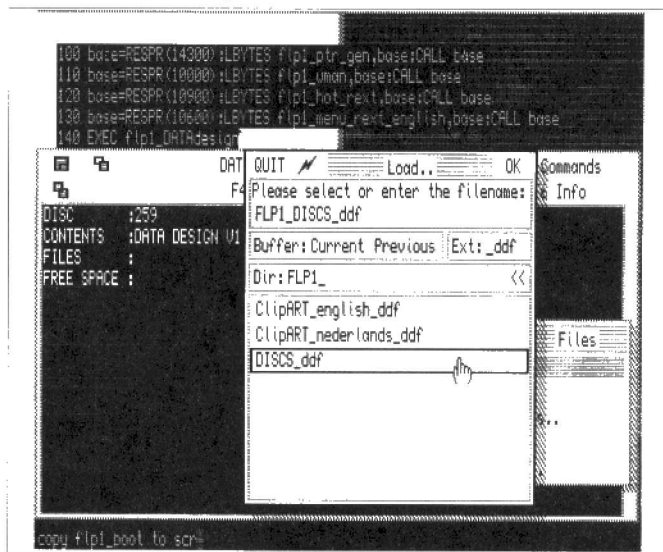
(larger than), < (smaller than), = (equal to) or NE (not equal to), to select records for printing. You can also specify the output device (default is "prt", equivalent to SER1; no others are listed), and set various print parameters, such as whether or not the field names should be printed, and what margins are to be used. The latter feature looked a distinct improvement over other programs, which usually assume you want "wall-to-wall" printing, but I failed to comprehend the mechanism for entering the margin settings and got garbage printed instead. In cases where the user

fied strings of characters, either in a particular field or in whole records, as requested; it is case-independent.

Sorting is done on one of two levels, in forward or reverse order, on any of the current field names, and on a numeric or alphanumeric basis. Sorting is switched off by default. There are no less than 11 boxes which can be clicked upon within this one sub-menu level, and the instructions do not cover their use adequately, but users should be able to figure out how to activate the basic sorting operations. The View command permits a particular field of all records to be displayed, in the sub-menu box. As a preliminary to printing, this allows the user to mark certain records only (using the cursor keys and Space bar) for printing. Alternatively, records can be marked for Deletion (from the current list). The Print operation can then be selected, and only the marked records will be printed. New Field allows the user to insert an additional field into an existing database; the new field is inserted as the last one of the record. Erase allows fields to be deleted, and all data contained within the specified fields will be deleted also. Truncate does not do what you might expect of it. As with some other usages in the menu system, the word chosen for the function is not the one many users will expect. In fact, Truncate is an "undo" or "undelete" operation; if a change is made within a record, and you decide not to implement it, Truncate will restore the record to its original condition. Delete is a standard operation, deleting the current record, if the action is confirmed as being required. There is a confirmation request when Quilt is selected also, if you have made changes and not Saved the database since then.

Extension

The second illustration shows the screen with the boot file copied to it; the SuperBasic command to do this is at bottom left, and the contents of the file across the top. The extensions files take up about 45 KB. The main Data Design window spreads across most of the screen width, with one file record being displayed at



pointer environment.

F2 is the key for file operations, and causes a menu to drop down, offering the options Load, Save, Save as, Forget and Print. The Load selection was referred to above. Save is chosen if the file has already been given a name, whereas Save As is used when a file is first named, or when it is to be renamed (ie making a backup copy). Forget removes the current database from the screen; more-familiar names for this option would be New, Clear or Zap. The Print option also appears after the View option of the F3 Commands menu, the difference being that it is assumed you want to print the whole file when the option is selected via F2, whereas the F3 route allows selection or deletion (inclusion/exclusion) of particular records. This is not to say that the F2 P keying restricts you to having all records printed, however. The Print menu allows you to enter a character string and a field, and use one of the operators >

has to enter parameters, a few examples of how to do this would be welcome.

The first illustration shows the main menu overlaid by the Files menu, itself overlaid by the Print menu; the latter has the = parameter selected, "SYSTEM" inserted as the search string and "CONTENTS" as the field in which to look for it, so that only records with the word SYSTEM in that field would be printed. Under Form, the "m 10" was intended to give a left print margin of 10 character-spaces, but it didn't work.

F3 is the key for commands, as it is with the Psion Quartet. The commands listed on the drop-down menu are Begin/first, End/last, Find, Sort, View, New field, Erase field, Truncate, Delete, and Quit program. The first two of these allow you to jump to the first or last record in a loaded database. A blank record is placed after the last used record, and this is where new data is entered. The Find function looks for speci-

the left. Above that is the menu panel; the two rectangular icons for moving and resizing the screen are at the left on the panel. Two levels of drop-down menu are superimposed on the main level; the one at bottom right appears when the F2 Files command is activated, and the larger one in the middle is produced by the Load command. Using the down cursor to move the pointer (which changes to a hand in this situation) has caused the required database file name to be highlighted by a box. Boxes above that reflect the current status – the files listed have the extension _DDF, the directed device is FLP1_, and the selected device and file are FLP1_DISCS_DDF. It would be helpful if the “f” in “filename” and the “D” in “Dir” were to be underlined, as the calling characters are on the previous menu level; pressing f puts the cursor into the filename box, so that a file name can be typed-in directly, and D does the same for the directory device box.

Samples

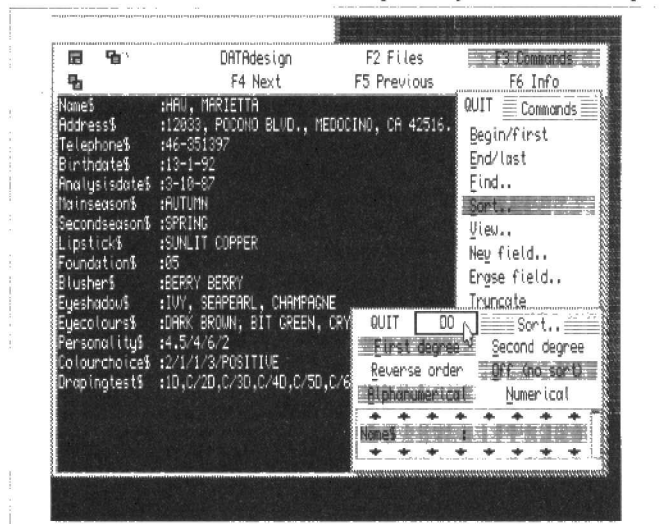
Database files are given the suffix _DDF. Two sample files are supplied. They are simple, two-field files with some rather obscure entries, but they suffice to show how to create records. Beneath the initial menu panel, there is a box for working in. To get into this, you press E, or move the pointer into it and press Space or ENTER. Unlike Archive, but similar to FlashBack, Data Design is free-form; you don't have a formal structure into which data is typed, but have what is more like a word-processing edit area to type field names and data into. Field names do not need to have the \$ ending, as used by Archive, but they can do so if imported from that program; a field is separated from its data by a colon. If one line is not enough to hold the data, pressing ENTER lets you use further lines. Even if you omit to make extra lines available this way, more data than the displayed line holds can be typed in; the front end of the line gradually disappears as you try to type beyond the trailing end, and all the line can be displayed by means of the left/right cursor keys, or ALT-left/right to go directly

to the line ends. Normal editing and deletion keyings can be used when entering data into fields.

When first creating field names, the down-cursor key is used to go to the next line, for entering the name of the next field. To get the set of field names accepted, you can press either F4 or F5, a colon being placed after each field name automatically. Once fields have been created, data can be entered in the usual fashion, with the TABULATE key being used to move between fields (with SHIFT, to move upwards). With data entered, the

correct interpretation. The sort order for alphanumeric sorting with the Sort and Print commands is first Space, then alpha characters in a-b order with capitals coming first (AaBb etc), then the numerals 0-9, then punctuation marks and special symbols with the comma and period first and the rest in the order of the character codes in the QL User Guide.

Archive Export files can be used in Data Design, after they have been modified by the CONVERT_BAS routine supplied. This has to be run separately from the main pro-



fields can be moved between with the up/down cursor keys, and the left/right keys move the cursor within the fields. You can't access the field name with the cursor. The normal next/previous record movement is by means of F4/F5.

The limitations placed upon field and record size are reasonable. Record length can be up to 4096 bytes (characters), field length up to 4095 if there is only one field per record, or 4096 minus the number of fields otherwise. The only limitation on number of records is the available memory to hold them, but no more than 65536 records can be sorted. The theoretical maximum number of (empty) fields is 4096, the number of lines per field is also 4096 (with 1 character per line). The number of fields which can be "controlled in Print" is '248 first fields'. I take this to mean that is the number of records that can be included/excluded on the View-Print menu, but this may be an in-

gram. Conversion takes a long time, for even a small file, but it will be a worthwhile function for users who have existing Archive databases they don't want to retype. As can be seen from the third illustration, the Archive field names are transferred 'as is' and the records are displayed properly; the example shown was used with a specially-created Screen in Archive, but the screen cannot be imported. The parameters of the Sort command are shown in the window overlaying the F3 Commands window.

A configuration program is supplied; this also has to be run separately, and the pointer environment has to be active beforehand. The colours can be chosen for the different areas of the program window, and the print device can be specified. The written instructions do not give guidance in changing the parameters, and there is no help on the screen either, so it's a case of suck-it-and-see here.

The speed of operation is good, and cursor movements are clean. Two databases con-

verted from Archive format were checked, and both looked quite correct. The largest file tested was about 200-300 records long, and there was no apparent change in speed from a minimal file of about ten records. Movement from one record to another is as fast as one could sensibly desire. The freedom of database structure and data entry is comparable to that found in FlashBack, which is a good standard to judge by. The overall 'front end' under which the program runs operates smoothly and is visually attractive, but is perhaps more appealing to users who have compatible mice than to keyboard-only users, as program control via the keyboard is long-winded. Program and instructions are heavily slanted towards the pointer environment – necessarily, because it controls everything. This makes the program very suitable for QRam/QPac users, but not so attractive for others.

Problems? By-and-large, nothing too serious. At one point, when F2 Files Load had been used several times in a row to get a satisfactory screen display to dump to the printer, something decided that no choice of file should be offered for loading, the test file being loaded every time (a useful function, if controllable!). Following this, the program refused to load anything and had to be run again. It seemed likely this problem was associated with my prolonged attempts to get the screen dump and would not occur in normal use. The Last Line Recall toolkit function disappeared. A system lockup occurs when a Find operation is requested and the current record is the last (blank) one. There was some difficulty sizing and moving the program window accurately, but this function is peripheral to the use of the program itself. It may be part of the same problem, that successive uses of a particular command (eg Find) are slowed-down by the pointer having moved away from the box that needs clicking upon (eg Do). The instructions are less than clear on certain functions, notably those involving the use of the pointer environment, but users would gradually figure out what they need to know.